The Dissertation Committee for Prapti Neupane
certifies that this is the approved version of the following dissertation:

Advances Towards a Multi-Dimensional Discontinuous Galerkin Method for Modeling Hurricane Storm Surge Induced Flooding in Coastal Watersheds

Committee:

Clint N. Dawson, Supervisor

Irene M. Gamba

Bjorn Engquist

Tan Bui-Thanh

Robert D. Moser

Advances Towards a Multi-Dimensional Discontinuous Galerkin Method for Modeling Hurricane Storm Surge Induced Flooding in Coastal Watersheds

by

Prapti Neupane, B.A.; M.S.C.S.E.M

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2016

Dedicated to my family.

Acknowledgments

Being a graduate student in the Computational Science, Engineering and Mathematics (CSEM) department at the University of Texas at Austin has been both fun and challenging. The coursework and the research have taken me out of my comfort zone and dropped me into territories I had not explored before. I would not have been able to complete this journey were it not for the encouragement, companionship and support I have received from the people that are in my life. Therefore, I would like to acknowledge and thank those people who have, in some way, contributed to the completion of this dissertation.

First, I would like to thank my adviser, Clint Dawson, for being a constant source of inspiration and motivation for me. His support and faith in me have helped me get over moments of self doubt, and his knowledge, foresight and guidance have helped me push through multiple hurdles I have encountered in my research. I would not have come to enjoy being a graduate student or living in Austin had I not had an adviser like him.

I am very thankful for having met some wonderful people in my first year in the CSEM program, with whom I got to share numerous terrible but wonderful grad school experiences like the rigors of our first year courses, the late night homework sessions and the terrors of prelims. I also want to ac-

knowledge everyone in the CHG group for all the feedback and input I have received on my work throughout the years. I want to thank the CSEM Fellowship and the National Science Foundation (NSF) for providing me the financial support during my graduate studies. I also want to thank the Institute for Computational Engineering and Sciences for introducing me to researchers and work being done at the frontier of all things computational.

Finally, I would like to thank my parents, my brother and my sister for always being there on the other side of the phone line, listening to my excited chatters or my frustrated ramblings, and for never wavering in their belief in me. I would like to thank my parents for instilling in me a love for education. I would especially like to thank my mom for teaching me to look up to and learn from people who have accomplished a lot in their lives and to not be discouraged by small failures. I am grateful to her for dreaming with me and walking with me every step of the way. This dissertation is as much her accomplishment as mine.

Advances Towards a Multi-Dimensional Discontinuous Galerkin Method for Modeling Hurricane Storm Surge Induced Flooding in Coastal Watersheds

Publication No. _____

Prapti Neupane, Ph.D.
The University of Texas at Austin, 2016

Supervisor: Clint N. Dawson

Coastal areas are regions of high population density and urbanization. These areas are highly vulnerable to inundation and flooding not only because of hurricane storm surge, but also because of the torrential rainfall that often accompanies hurricanes. In order to accurately predict the extent of damage such an event might cause, any model that is used to simulate this process needs to couple rainfall with storm surge. The works that have tried to address this issue have mostly used a unidirectional coupling technique, where one of the following two approaches is taken. In the first approach, a hydrology model is used in the domain of interest, and storm surge is incorporated in the domain as a boundary condition. In the second approach, a storm surge model is used in the domain of interest, and rainfall is incorporated in the domain as a river inflow boundary condition. Neither of these approaches allows the rainwater and surge water to interact bidirectionally.

In order to improve on those efforts, in this dissertation, we develop a comprehensive framework for modeling flooding in coastal watersheds. We present an approach to decompose a watershed into multiple sub-domains depending on the dynamics of flow in the region. We use different simplifications of the shallow water equations on different sub-domains to gain computational efficiency without compromising on physical accuracy. The different sub-domains are coupled with each other through numerical fluxes in a discontinuous Galerkin framework. This technique allows for a tight coupling of storm surge with rainfall runoff, so that the flooding that occurs is truly influenced by the nonlinear interaction of these two processes. We present numerical tests to validate and verify the methods used for modeling flow in different sub-domains as well as the techniques used for coupling different sub-domains with each other.

Table of Contents

List of Figures

Chapter 1

Introduction

Coastal regions are areas of high population density and urbanization. About half of the world's population lives within 200 kilometers of coastlines and 14 of the world's 17 largest cities are located along coasts [24]. These areas are highly vulnerable to inundation and flooding not only because of hurricane storm surge, but also due to the torrential rainfall that often accompanies hurricanes. This vulnerability to flooding was realized in south Texas and northeastern Mexico after Hurricane Alex, and subsequently in northeastern United States after Hurricane Irene. Hurricane Alex made landfall in 2010 about 100 miles south of the Rio Grande, dumping heavy rainfall to the region. To prevent overflow of water from dams and reservoirs, emergency water managers released large amount of this water into adjacent floodways. However, eight days after Hurricane Alex made landfall, before the surge and the rain water had a chance to diminish, a tropical depression brought more rain to the region. This rainfall caused severe flooding and damaged thousands of homes [84]. In 2011, Hurricane Irene made landfall in North Carolina, New Jersey, New York and Connecticut. The resulting winds and flooding killed 67 people and cost $15.8 billion in damages. Much of the flooding had happened as a result of the heavy rainfall that accompanied the hurricane and the

surface runoff that it caused [3].

The examples cited above demonstrate the need for accurately predicting how surge and flood water, collectively referred here as storm water, propagate and are routed in coastal areas. Being able to anticipate and predict the extent of flooding during and after a hurricane is important for emergency management planning to protect the coastal population and infrastructures. We have tried to address this issue in this dissertation by developing a comprehensive framework for modeling flooding in coastal watersheds. A watershed or drainage basin is defined as a geographic extent that shares a common point, called a sink, to which all surface water, for example from storm surge or rain, drains. The surface water can be carried to the sink via a combination of surface runoff flow, also called overland flow, and free surface flow occurring in natural rivers and streams as well as man-made channels.

A typical domain for simulating hurricane induced flooding in coastal watersheds consists of the following three regions: i) the coast and a small portion of a bay or an ocean near the coast, ii) hillslopes/floodplains that receive rainfall and produce runoff, and iii) networks of small streams and channels that receive rainfall runoff. The physical characteristics (e.g., infiltration rates, friction coefficients, etc) and the geometrical characteristics (e.g., bathymetry and topography) of these regions vary over a wide range of length scales. These characteristics play a significant role in determining the characteristics of flow in these regions. Although typical meshes used for running such simulations use elements of different sizes in different regions to capture their length scales,

a computationally feasible mesh is generally not able to accurately resolve the length scales of small channels. However, adequate representation and resolution of small-scale channels is essential for making accurate flood predictions because routing of surge and flood water in watersheds is largely determined by these channels [26].

To tackle the aforementioned complexities, we have developed a multi-dimensional modeling approach that will resolve the multiple scales present in a coastal watershed and preserve physical accuracy while being computationally efficient. In addition, our approach truly couples storm surge with rainfall runoff dynamically in time, allowing rainfall runoff as well as storm surge to contribute to the flooding of a watershed. This coupling is done via numerical fluxes in a Runge–Kutta discontinuous Galerkin (RKDG) framework. Just like finite volume methods, these methods are particularly suited for solving hyperbolic problems. These methods can naturally have a high order of accuracy, allowing us to achieve a higher spatial and temporal accuracy than finite volume methods. Additionally, these methods are robust and are well suited for solving problems on unstructured grids. This is important because the domains on which we want to run flooding simulations very commonly have irregular boundaries.

1.1 Background

In this section we will discuss different mathematical models that are used to represent the hydrological processes of a watershed. Then, we will

review popular softwares and techniques for distributed hydrological modeling of watersheds. We will also review various hydrodynamic models and softwares that are used to model storm surge in coastal areas. We will then discuss how in the recent works hydrologic models/softwares are used along with storm surge models/softwares to model flooding in coastal areas. Finally, since we are using a RKDG method in our work to solve hyperbolic partial differential equations, we will conclude this section with a review of the development of and a discussion of RKDG methods for the solution of hyperbolic partial differential equations.

1.1.1 Modeling of Watershed Hydrology

The USGS defines hydrology as the "science that encompasses the occurrence, distribution, movement and properties of the waters of the earth and their relationship with the environment within each phase of the hydrologic cycle" [80]. Watershed hydrology is defined as "that branch of hydrology that deals with the integration of hydrologic processes at the watershed scale to determine the watershed response" [87].

Various methods have been developed to study hydrologic processes in a watershed. The unit hydrograph method was developed by Sherman [85] for describing the relationship between rainfall excess and direct runoff. The unit hydrograph of a watershed is defined as "a direct runoff hydrograph resulting from 1 inch of excess rainfall generated uniformly over the drainage area at a constant rate for an effective duration" [15]. It is built on the assumption that

watersheds behave as linear systems. However, hydrologic systems are usually nonlinear because of factors such as storm origin and patterns and stream channel hydraulic properties. The unit hydrograph is only applicable when channel conditions remain unchanged and watersheds do not have considerable storage, which is clearly violated when channels flood and create considerable storage in floodplains.

There are two categories of models used to handle flooding situations in watersheds viz. lumped and distributed flood routing models. In a lumped model, for a hydrologic system, input $I(t)$, output $Q(t)$, and storage $S(t)$ are related by the continuity equation:

$$\frac{\partial S}{\partial t} = I(t) - Q(t) \tag{1.1}$$

We know the input $I(t)$ to the watershed, and we are interested in solving for $S(t)$ and $Q(t)$. Therefore, a second relationship called the storage function, relating S, I and Q is needed to close the system. This relationship can be written as:

$$S = f\left(I, \frac{\partial I}{\partial t}, \frac{d^2 I}{dt^2}, \ldots, Q, \frac{\partial Q}{\partial t}, \frac{d^2 Q}{dt^2}, \ldots\right) \tag{1.2}$$

The exact form of this relationship is what distinguishes different lumped models from one another. A very commonly used model for the storage function is the Muskingum method given by:

$$S = K[XI + (1 - X)Q] \tag{1.3}$$

where, K is a proportionality coefficient, and $X \in (0, 0.5)$ is a weighting factor. A drawback of lumped models is their inability to allow for spatial variability

of flow rates and water level throughout a watershed. A second disadvantage of lumped models is that the calculation of water level has to be done separately from the flow rate calculation by assuming steady nonuniform flow along the channel at the desired location. However, in order to simulate actual unsteady nonuniform nature of flow propagation in channels, flow rates and water levels have to be computed simultaneously. These two drawbacks of lumped flow routing models can be mitigated by using distributed flow routing models.

Distributed flow routing models are based on partial differential equations that allow the flow rate to be computed as a function of space and time. The Saint-Venant equations, developed by Barre de Saint-Venant in 1871, belong to this category of distributed hydrologic models. These equations are used for describing one-dimensional unsteady open channel flow and can be written in the following form by neglecting lateral inflow, wind shear and eddy losses:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0 \tag{1.4}$$

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right) + gA\frac{\partial H}{\partial x} - gA(S_0 - S_f) = 0 \tag{1.5}$$

In the equations above, A stands for the cross-sectional area of the flow, Q is the volumetric discharge rate, H is the water height, S_0 is the bed slope and S_f is the friction slope. Equation (1.4) is called the continuity equation and it expresses the principle of conservation of mass. Equation (1.5) is called the momentum equation and it expresses the momentum conservation principle. When all the terms in (1.5) are retained, the hydrologic model is referred to as

the *dynamic wave model*. When the inertial forces, i.e. the local acceleration term (the first term in (1.5)) and the convective acceleration term (the second term in (1.5)) are neglected, we obtain the *diffusion wave model*:

$$g\frac{\partial H}{\partial x} - g(S_0 - S_f) = 0 \tag{1.6}$$

When the pressure term (the third term in (1.5)) is neglected in addition to the inertial forces, we obtain the *kinematic wave model*:

$$S_0 = S_f \tag{1.7}$$

This equation implies a balance between the gravitational and the friction forces.

1.1.2 Popular Techniques in Distributed Hydrological Modeling

Flooding is very commonly modeled as a one-dimensional unsteady flow using either the 1-D Saint Venant equations, also called the section-averaged shallow water equations, or some other simplified version of those equations [37]. In this approach, the entire simulation domain, consisting of floodplains and narrow channels and rivers, is represented with a 1-D network that is composed of linked 1-D sections [79]. Widely popular commercial softwares like MIKE11 [43], developed by DHI (formerly known as the Danish Hydraulic Institute), as well as open source softwares like HEC-RAS (Hydrologic Engineering Center's River Analysis System) [10], developed by the U.S. Army Corps of Engineers, have adopted this approach. These softwares are fast, owing to the 1-D nature of their models. However, the drawback is

that these models do no always accurately represent the flooding that occurs once the channels and rivers have reached their capacity [60]. Flooding on land is a two-dimensional process and models that can handle lateral diffusion of the flood wave and backwater effects need to be used to capture this phenomenon [66, 83].

Two-dimensional models provide a more accurate description of flow. They allow the flow velocity to have two dimensional components, thus eliminating the need to predefine flow paths. These models use the 2-D Saint Venant equations, also called the 2-D shallow water equations, as the governing equations. Flood modeling softwares like MIKE21 [2], and FLO-2D [1] use this approach. However, resolving all the small scale features with a 2-D mesh requires a very fine mesh, which makes the simulation computationally intractable, and often data on model parameters, like bathymetry, is not available at such a fine resolution.

A natural way to deal with the deficiencies of the 1-D models and the 2-D models is a multi-dimensional approach, where small-scale features like channels and rivers are modeled as 1-D domains, and flood plains are modeled as 2-D domains. Recently, such 1-D/2-D coupled models have been gaining a lot of attention in the flood modeling community. These models are attractive because, while they reduce the computational cost associated with a fully 2-D model, they represent the physics more accurately in regions where a 2-D model is essential. We will briefly review a couple of such models here.

Kuiry, Sen and Bates [57] developed a 1-D/Quasi 2-D model by cou-

pling a 1-D river flow model with a simplified storage cell floodplain flow model. They solve the full 1-D Saint Venant equations on the river using a finite volume scheme. The floodplains are discretized using an unstructured triangular grid where a mass continuity equation is used to relate the net flow into a cell with its corresponding change in volume [57]. This technique does not allow for the calculation of flow velocity in the floodplains. Seyoum et al. [83] developed a slightly more complicated 1-D/2-D model that overcomes this deficiency. They solve the 1-D section averaged Saint Venant equations on the 1-D domains. The 2-D domains are discretized using a regular rectangular grid, and a non-inertial simplification of the 2-D Saint Venant equations are solved on these domains. SOBEK 1D2D, a commercial software developed by Deltares (a Dutch applied research insititute), is more sophisticated than either of these two models. This code solves the dominant main channel as a 1-D structure and the floodplains as 2-D domains, and in these domains, either the 1-D or the 2-D Saint Venant equations are solved. It uses a structured grid for the 2-D domain and solves the 1-D and the 2-D systems with a staggered finite volume scheme [94]. MIKE-FLOOD, developed by DHI, is another commercial software that uses a coupled 1-D/2-D approach for modeling flooding. It solves the same set of equations as SOBEK 1D2D. The equations are solved on staggered C-grid with a semi-implicit ADI (Alternating Direction Implicit) two-step finite difference algorithm [79].

While the models mentioned above are improvements on the 1-D network models, the most glaring of their shortcomings is in their inability to

use an unstructured grid. Most watersheds have complex geometries that can be adequately represented only with an unstructured grid. The new release of HEC-RAS aims to overcome this problem [11]. It incorporates a combined 1-D and 2-D modeling framework. 1-D channels are represented by linked 1-D segments, and the section-averaged Saint Venant equations are solved on these segments. The 2-D floodplains are represented by an unstructured grid, where either the full 2-D Saint Venant equations or the diffusion wave equations are solved. The equations are all solved using finite volume methods [11]. The hydrology portion of our modeling framework improves on these models in the following way. The floodplains will not be statically classified as 2-D domains. We will dynamically classify the floodplains as either 1-D or 2-D domains, depending on the characteristics of the flow. Depending on the flow conditions in a particular domain, we will use different sets of governing equations like the 1-D and the 2-D shallow water equations and the kinematic wave equation. These equations have varying orders of computational complexities. Thus, by using this multi-dimensional and multi-physics approach, we avoid the computational cost that would be incurred if the floodplains were statically represented as 2-D domains governed by the full 2-D Saint Venant equations.

1.1.3 Hydrodynamic Modeling of Storm Surge

Storm surge is a "meteorologically forced long wave motion, which can produce sustained elevations of the water surface above the levels caused by the normal astronomical tides" [7]. The primary factors driving storm surge are

extreme wind stress from the storm, bottom stress and reduced atmospheric pressure to a lesser extent [7, 26]. A wave with wavelength substantially larger than its amplitude is called a long wave. The characteristic time-scale of storm surges range from several hours to one day and a wavelength is approximately equal to the width of the center of the depression typically between 150 and 800 km [72]. Storm surge is generally modeled using shallow water equations. These equations can be derived from the Navier-Stokes equations for a turbulent fluid using Reynolds averaging to describe the mean flow. They are derived under the assumption of a hydrostatic pressure and the Boussinesq approximation for long waves. The hydrostatic assumption makes the vertical acceleration and associated viscous terms in the vertical momentum equation negligible. The Boussinesq approximation allows gradients of fluid density to be ignored [30].

3-D models are attractive because of their ability to provide information on the vertical structure of storm-induced currents [7]. However, these equations can be computationally expensive to solve. Since storms induce significant mixing throughout the water column near shore, it is sufficient to use the 2-D shallow water equations in nearshore regions [26]. The 2-D shallow water equations or the depth-averaged shallow water equations can be derived by integrating the 3-D shallow water equations in the vertical direction. If the area of interest covers a large area of the Earth's surface, then these equations need to be expressed in spherical coordinate system rotating with the earth. However, for the purposes of this dissertation, we are only interested

in representing a small portion of an ocean. Therefore, we will limit ourselves to Cartesian coordinates. We will present these equations in a subsequent chapter.

Below, we will briefly review four softwares that are popularly used to model storm surge.

SLOSH: Much of the information presented here is summarized from the National Hurricane Center website [69] and the NOAA technical report about SLOSH [47]. SLOSH is an acronym for Sea, Lake and Overland Surges from Hurricanes, and is a numerical model developed by the National Weather Service (NWS) to estimate storm surge heights resulting from historical, hypothetical or predicted hurricanes. It is run by the National Hurricane Center to forecast storm surge in real time and also by the Federal Emergency Management Administration (FEMA), the U.S. Army Corps of Engineers, and state and local emergency managers as a tool for hurricane evacuation planning [41]. In the model, storm surge is driven by a wind field generated using the atmospheric pressure, size, forward speed and track data of the hurricane. The governing equations used in the model are a linearized form of the two-dimensional shallow water equations with the advective terms ignored. These equations are numerically solved using a finite difference scheme on a continuously varying polar grid system. The choice of the grid system allows for a fine mesh in the limited area nearest the pole point, stretching continuously to a coarse mesh at distant boundaries of a large basin. Thus the simulation

domain can be large without substantially incurring a big computational cost. This approach avoids the problems associated with using a small domain for the simulation and prescribing a boundary condition in regions of significant surge activity, which is often inaccurate. The model does not account for normal river flow and rain. However, river flow upstream can be incorporated as a boundary condition, and rain can be incorporated as a source term. The model is not coupled with astronomical waves even though it is well known that these waves can significantly influence the solution [7].

Japan Meteorological Agency (JMA) Storm Surge Model: The JMA Storm Surge Model has been utilized by Japan Meteorological Agency to provide storm surge information since 1998. This model solves the two dimensional shallow water equations using an explicit finite difference method. Similarly to SLOSH, the non-linear advection terms in this model are omitted for computational efficiency.

ADCIRC: The ADvanced CIRCulation model (ADCIRC) is a finite element based code that uses unstructured, triangular grids, enabling a highly accurate representation of irregular coastlines. By using elements of different sizes, ADCIRC has the ability to use basin scale domains for a storm surge simulation without significant computational cost [30]. It can be run either as a 2-D depth integrated model or as a 3-D model. The 2-D and the 3-D models solve a variation of the shallow water equations, where the primitive continuity

equation is replaced with the generalized wave continuity equation (GWCE) in order to avoid spurious modes without adding artificial diffusion to the numerical scheme, which would significantly compromise the solution [61]. The 2-D and 3-D velocities are obtained by solving the momentum equations in non-conservative form. Storm surge is driven by surface wind stresses, tidal potential and atmospheric pressure gradients. Rainfall runoff can be incorporated into the model as a river boundary condition. ADCIRC has also been coupled with a wave model called SWAN (Simulating WAves NearShore) to allow for interaction of short wind-driven waves with the long waves of circulation [29]. ADCIRC has been validated by hindcast studies of different hurricanes like Hurricane Katrina and Hurricane Rita [13, 28], Hurricane Gustav [32] and Hurricane Ike [45].

DG-SWEM: The Discontinuous Galerkin Shallow Water Equations Model (DG-SWEM) solves the depth averaged shallow water equations in conservative form using a Runge–Kutta discontinuous Galerkin method or a Runge–Kutta local discontinuous Galerkin method [26, 55]. This model is similar to ADCIRC in many ways. It can be run with the unstructured triangular meshes developed for ADCIRC. The effects of hurricanes and rainfall runoff can be incorporated into a simulation domain in a manner similar to ADCIRC. This model is preferable to ADCIRC because it preserves local mass conservation. Furthermore, discontinuous Galerkin methods are better suited for advection dominated flows than finite element methods. In ADCIRC, the advection of

eddies can be resolved either by using a very fine mesh or by the addition of eddy viscosity. In contrast, DG-SWEM is able to resolve such advection on a much coarser grid and without the use of any eddy viscosity or slope limiters [55]. A common cited disadvantage is the increase in the number of degrees of freedom in DG-SWEM as compared to ADCIRC and the increase in runtime as a result. However, for the same computational cost, DG methods can achieve higher accuracy than CG (continuous Galerkin) methods and in the DG framework, better scaling can be achieved on large-scale parallel machines because of the ease with which elements can be decoupled [56]. Furthermore, this cost can also be offset by using higher order elements, which will decrease the ratio of the repeated degrees of freedom to the total number of degrees of freedom, and will result in higher accuracy [55]. The validity of this model has been established by simulating Hurricane Ike and comparing the result with the one obtained by using ADCIRC [26].

1.1.4 Review of Coastal Flood Models

There are two popular approaches to modeling flooding in coastal watersheds. The first approach uses either a hydrologic or a hydrodynamic model in the watershed. In this category, very sophisticated models that accurately represent the geometry of rivers and channels as well as floodplains have been developed. HEC-RAS, Vflo, FLO-2D and LISPFLOOD-FP are the most commonly used softwares in this category. The domain for these simulations typically include hillslopes, floodplains, rivers, drainage networks and coastal

areas. The boundary of these domains are drawn at the coastline because the hydrodynamics in these models are often simplified and might break in oceans [6]. Simulations on such domains incorporate the effects of storm surge by prescribing either a time-varying mass flux (discharge) or a time-varying water surface elevation (stage) as boundary conditions on the boundary along the coastline.

Bates et al. [6] simulated a real flooding event that happened in East Anglia, UK, in 1953 using LISPFLOOD-FP. This work incorporated storm surge into the simulation by using the recorded time series of water levels at the coastline as boundary condition. Lapidez et al. [58] used FLO-2D to determine areas vulnerable to flooding because of storm surge in the Philippines. In this work, JMA Storm Surge Model was used to generate storm surge due to hypothetical typhoons, and the water surface elevation time series obtained using the JMA simulation and the astronomical tide levels from WXTide were used to create a stage-time relationship that was then provided as boundary condition to FLO-2D. Brown et al. [9] used a more dynamic approach to simulate flooding induced by storm surge. They used a flood model in regions farther away from the coast and used a storm surge model near the coast. They dynamically coupled their flood model with their storm surge model in such a way that flow from the storm surge model was allowed into the flood model and flow was allowed out of the flood model. However, the storm surge model was not influenced by the flood model. Similarly, in [74], HEC-HMS, a lumped hydrologic modeling system designed by the HEC, was used in conjunction

with HEC-RAS to simulate the effects of rainfall in the Houston-Galveston area watershed during storm events. Effects of storm surge were incorporated in the domain by using the stage hydrographs generated by ADCIRC as downstream boundary conditions on the HEC-RAS domain. Therefore, this work was also not a true two-way coupling between the storm surge domain and the hydrology domain.

The second approach uses a hydrodynamic model in the coastal area. The domain of such simulation typically includes parts of an ocean or bay and land area close to the coast. Westerink et al. [95] used ADCIRC to simulate hurricane storm surge in southern Louisiana. They validated the ADCIRC model through hindcasts of Hurricanes Betsy and Andrew. However, their simulation did not directly take rainfall runoff into account; Rainfall runoff was incorporated in the simulation through river boundaries where a radiation boundary condition, that specified flux into the domain while allowing surface waves to propagate out, was imposed. A similar approach was taken by Bunya et al. [13] for hindcasting Hurricanes Katrina and Rita. While they achieved good results by coupling kinematic winds from observational data, deep-water wind waves generated by the Wave Model (WAM) and nearshore waves generated using Steady-State Irregular Wave (STWAVE) with ADCIRC for predicting water elevation and currents during the hurricanes, their model was not coupled with any watershed hydrologic model. Riverine flow was incorporated as boundary condition. Citing that applying river inflows to storm surge models as constant values representative of flow rates during landfall

is not adequate for storm events with a significant precipitation component, Tromble et al. [90, 91] coupled HL-RDHM and Vflo, two watershed hydrologic models, to ADCIRC. Although this coupling was done dynamically in time, the coupling was not a two-way coupling. Information was exchanged at a "handoff" point, where information from the hydrologic models was passed on to the storm surge model, but the hydrologic model did not receive any information from the storm surge model. Therefore, there was no mechanism for providing the hydrologic model with the knowledge of ocean boundary condition. Realizing the limitation of their approach, the authors noted that when the "handoff" point was too far downstream, the solution showed significant degradation, and thus they avoided this problem by extending the storm surge domain sufficiently further inland.

The approaches described above either treat storm surge and rainfall as separate problems or couple them only very loosely. This approach is problematic because when rainfall runoff moving through a channel towards the ocean meets storm surge moving inland through the same channel, any flooding that would normally occur is exacerbated [74]. However, in the lack of a bidirectional coupling between storm surge and rainfall, this phenomenon cannot be adequately captured. We mitigate this problem in this dissertation by developing a true bidirectional approach for coupling channels with the storm surge domain.

1.1.5 Review of Runge–Kutta Discontinuous Galerkin Methods

The original discontinuous Galerkin finite element method (DGFEM) was developed by Reed and Hill in 1973 for solving a time-independent neutron transport equation [75]. Following its introduction, different flavors of discontinuous Galerkin methods were developed for use in solving ODEs as well as time-dependent PDEs. For solving transient PDEs, some methods used discontinuous Galerkin discretization in both time as well as space. Others used an implicit discretization scheme in time in conjunction with a discontinuous Galerkin discretization in space. See [21] for a detailed historical review of the development of DG methods.

The explicit RKDG method uses the explicit stable Runge–Kutta time-stepping scheme along with the discontinuous Galerkin discretization in space. It was developed by Cockburn and Shu in a series of papers for solving one- and multi-dimensional scalar as well as systems of conservation laws in the late 80s through the 90s [16–20]. This method was first applied by Schwanenberg and Kongeter [82] in 2000 to the planar one-dimensional shallow water equations. Since then, there has been a proliferation of literature on the application of RKDG methods to the shallow water equations (e.g. [5, 8, 31, 38–40, 42, 52–54, 63, 81, 98]).

Since the shallow water equations are used to model flow in coastal regions, any numerical method that attempts to solve these equations needs to have these two physical features: the well-balanced or exact C-property, and the ability to handle the wetting and drying interface correctly. These prop-

erties have been and continue to be extensively researched. Well-balancedness means that the numerical method is able to preserve still water at rest. This property can be achieved in an RKDG framework by approximating the bathymetry or the topography terms in the same finite element space where the primitive variables are approximated [52, 97]. This is straightforward in the case of low-order polynomial approximations, but is more involved while using high-order polynomial approximations [50, 52, 98]. In regards to wetting and drying, there are two major ways it is handled in the literature. The first method is a mesh modification method, where, either the wet/dry interface is treated as a moving boundary problem and the mesh is modified accordingly, or some elements in the mesh are removed from the solution as they get dry and are added back if they get wet. The technique used by Bokhove [8] falls into this category. The second method uses a thin water layer approach, where a minimum water height is maintained at all the computational nodes by flux modification and/or post-processing of the solution. This approach seems to be more common in the literature [12, 42, 50].

Since flood modeling softwares are often used for predicting flooding and making evacuation decisions, other features that a numerical method that solves the shallow water equations needs to have are numerical accuracy and computational efficiency. To this end, Giraldo and Warburton [39] investigated the solution of the *strong* form or *divergence* form of the shallow water equations as opposed to the most commonly used *weak* form or *Green's* form, using high-order nodal polynomial approximations of the primitive variables. Ku-

batko, Westerink and Dawson [53] investigated the solution of the *weak* form of the shallow water equations using high-order modal polynomials. The *hp* convergence properties of the RKDG methods with regards to the shallow water equations have also been studied. It has been shown numerically that just like continuous Galerkin (CG) methods, RKDG methods also show optimal convergence – $\mathcal{O}(h^{p+1})$ – i.e. convergence rates of $p+1$ in h and exponential convergence in p [31, 53]. Thus, these methods achieve high accuracy. Current direction of research in the RKDG-for-shallow-water-equations community is towards developing fast algorithms and softwares, either by leveraging the ease with which parallel implementation and adaptive *hp* refinement can be achieved in an RKDG framework [54, 76] or by accelerating the computation with external devices like GPU [34].

Along with the important features discussed above, RKDG methods also possess the following properties: i) local conservation of mass and momentum, ii) efficient and stable reproduction of shocks and discontinuities, and iii) geometric flexibility to use unstructured grids. Because of these properties, RKDG methods are ideal for simulation of surface flow in a watershed where accuracy and speed are crucial.

1.2 Outline

The rest of this dissertation is organized in the following manner. In Chapter 2, we discuss the framework we use to model flooding in coastal watersheds. This includes description of the decomposition of the entire simulation

domain into smaller sub-domains that differ from each other in terms of the flow dynamics. We also introduce the set of governing equations that model flow in each of these sub-domains. In Chapter 3 and Chapter 4, we describe the solution of the governing equations in two of the sub-domains. We present different numerical examples to validate the models and verify the numerical technique used to solve the governing equations. In Chapter 5, we detail the interface conditions that are used to couple various sub-domains with one another. We present numerical examples to show that the coupling strategy produces viable results. Chapter 6 is the culmination of all the developmental work we have presented in the preceding chapters. Here, we present the results of simulating a flooding event in an artificial watershed that has all the sub-domains that typically exist in domains used for coastal flood simulations. Finally, in Chapter 7, we summarize the work we have done and provide an outlook towards the future.

Chapter 2

Modeling Framework

The hydrologic process we are interested in modeling begins either with rainfall or with inflow of surge water into a watershed or a combination of both. Some of this water is absorbed into the land via infiltration; the rest gets routed over land surfaces as overland flow and through networks of streams and channels as free-surface or open-channel flow. Eventually all of this water is carried to a sink. There are two major length scales involved in this process – the length scales of overland flow surfaces and that of the small streams and channels. Overland flow surfaces have larger length scales compared to that of the small streams and channels. Thus, a relatively coarse mesh can resolve the spatial features of overland flow surfaces, whereas, a very fine mesh is needed to resolve the spatial features of small streams and channels with the same accuracy. For flow in regions with small length scales, advective forces are dominant, whereas for flow with large length scales, bed friction will dominate and the advective forces can be neglected [46]. Thus, our domain of interest contains regions that not only need different mesh resolutions, but also have different kinds of flow that that are usually modeled by different physical equations. We resolve the complications arising from such multiple scales and multiple types of flow by using the dynamic domain decomposition

technique described below.

2.1 Domain Discretization and Decomposition

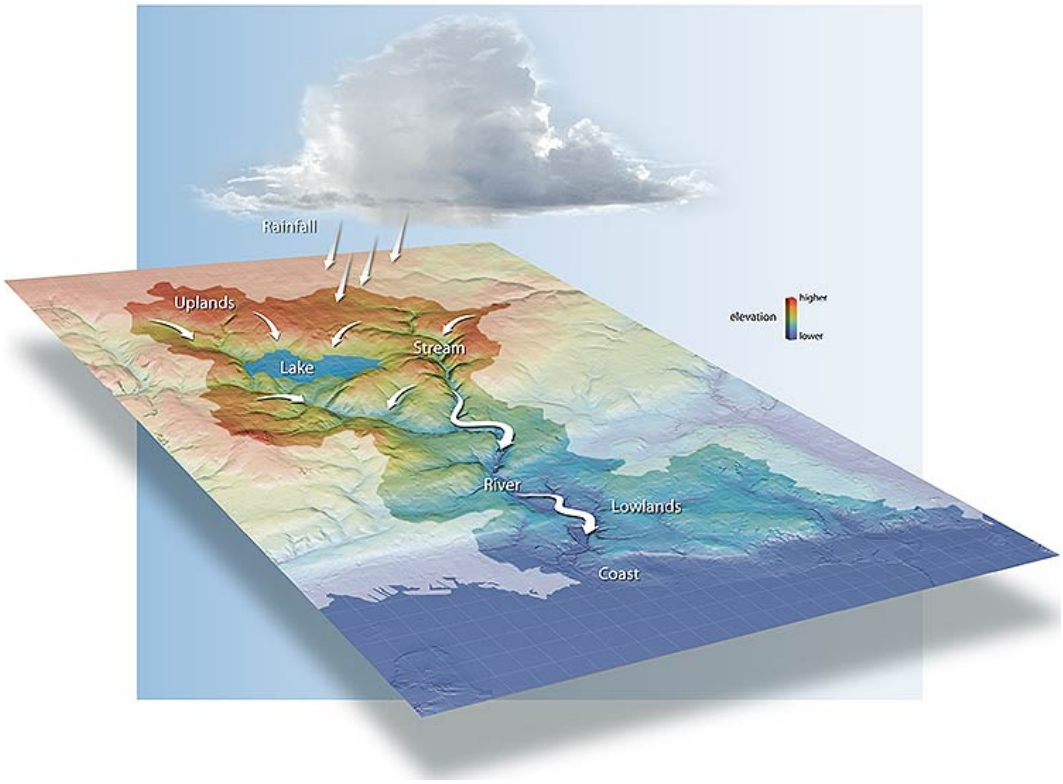


Figure 2.1: An illustration of a typical coastal watershed and the domain decomposition[1]

In Figure 2.1, we have illustrated what a typical coastal watershed might look like. We discretize such a watershed using an unstructured mesh

[1]Source: https://www.swfwmd.state.fl.us/education/watersheds/

consisting of triangular elements. Once the domain has been discretized, we decompose the domain into three sub-domains. The three sub-domains, as illustrated in Figure 2.1 are: channel network, overland flow region and shallow water region. Below we will briefly discuss each of these sub-domains.

I. Channel network:

Natural watersheds have a complex network of channels and streams. However, the free surface flow equations that are used to model flow in such channels and streams are only valid for a single channel. Traditionally, this problem is overcome by splitting a network of channels into multiple channels that are coupled together by prescribing some interior boundary conditions at the junctions. The interior boundary conditions that are inexpensive to impose are too simple to capture the physics at the junctions, while the ones that capture the physics more accurately are prescribed via solutions to non-linear equations, which can be expensive to solve. Additionally, this approach does not always capture the 2-D effects that are introduced at the junction because of flow coming into a junction from multiple channels at different velocities. Therefore, we decompose our channel network sub-domain further into channels and junctions.

The channels sub-domain represents channels that are too small to be adequately resolved by a 2-D mesh. In cases where the mesh is not fine enough to resolve the spatial variation of the domain, using higher dimensional equations to model flow properties will not improve the accuracy of the flow prop-

erties calculated. Instead, these equations may introduce unnecessary noise in the solution. In fact, for narrow channels and streams, 2-D models solved on a computationally feasible 2-D mesh will result in less accurate results than 1-D models solved on 1-D meshes that provide a detailed description of the cross-sections. Therefore, in our model, we represent such channels as being 1-D domains. In a mesh, 1-D channel segments are represented by a collection of edges of triangular elements that discretize another sub-domain called the overland flow region. Flows in channels can have large depths and high velocities. Waves in channels can travel in downstream as well as upstream directions. A good model will have to be able to reproduce these effects, and for this reason, we use the 1-D Saint Venant equations to model flow in this sub-domain. These equations will be introduced in Section 2.2.

The junction sub-domain is represented as a 2-D domain. This sub-domain facilitates coupling of channels with one another. We model flow in junctions using the 2-D shallow water equations. These equations will be introduced in Section 2.2.

II. Overland flow region:

This sub-domain represents land surfaces that are initially dry. This domain receives rain water which flows down its surface as overland flow or surface runoff. Overland flow is driven by the slope of the surface. It is a common practice to model such flow as kinematic flow.

Let us introduce two dimensionless quantities that indicate when kine-

matic flow is a good approximation. The first dimensionless quantity is the Froude number (F). Froude number is the ratio of inertial to gravitational forces, and is given by: $F = \frac{v}{gH}$, where v is the velocity of flow, g is the acceleration due to gravity and H is the water height. The second dimensionless quantity, called the kinematic wave number or the kinematic flow number (K), reflects the effects of the length and the slope of flow surfaces. Kinematic wave number is given by: $K = \frac{S_0 L_0 g}{H_0 F_0^2}$, where S_0 is the slope of the plane, L_0 is the length of the plane, H_0 is the normal depth at $x = L_0$ and F_0 is the Froude number of normal flow at $x = L_0$ at a discharge of $q_{max} L_0$, where q_{max} is the maximum rate of lateral inflow (see [96]). Normal depth is the depth of flow in a channel or culvert when the slope of the water surface and channel bottom is the same resulting in a constant water depth.

When the Froude number is less than 1.5, dynamic waves are dampened out and kinematic waves dominate in the flow [68]. Kinematic wave theory is accurate when $K > 20$ and for flows in which the condition, $KF_0^2 \geq 5$, holds [88]. Overland flows are subcritical, i.e. their Froude number is less than 1, and have kinematic wave numbers greater than 50, making kinematic wave a good model for such flows [88]. The kinematic assumption is violated for very flat ($S_0 < 0.002$) or for very steep ($S_0 > 0.10$) land surfaces [68]. However, the land surfaces over which we want to model the flow will have moderate slopes, well within the range for which the kinematic wave approximation is valid. Because of these considerations and the simplicity of the kinematic wave model, we use this approximation to model flow in the overland flow

region. The kinematic wave equation is a one-dimensional equation. Therefore, although overland flow region is discretized with triangular elements, making the domain a two-dimensional domain, flow calculations in this sub-domain will be one-dimensional. The kinematic wave equation will be described in detail in Section 2.2.

III. Shallow water region:

This sub-domain represents the part of the domain that is inundated by water. In a coastal watershed, this sub-domain might represent part of an ocean or a bay, or the land region along the coast that receives surge water, or the river and channel banks that have been flooded. Unlike surface runoff, these flows do not necessarily occur in the direction of the surface slope. Often, they can travel upstream against gravity. They can spread out as two-dimensional flows. Thus, we model such flows with the 2-D shallow water equations.

The classification of a sub-domain as overland flow region or shallow water region is dynamic in time. At every step of the simulation, we evaluate the channels to determine if any of them have flooded. If a certain channel has flooded, floodwater needs to be routed out of the channel on to the floodplains. However, overland flow regions will not be able to handle the dynamics of a flood wave. This is because the kinematic wave equation we use to model flow in these regions is a one-dimensional hyperbolic scalar balance law. It has one characteristic, the slope of which is the velocity of wave propagation. Thus, this equation allows waves to travel in only one direction. However,

flood waves propagate longitudinally as well as laterally. To accommodate this process, we convert the overland flow region into a shallow water region. Once the floodwater recedes, the shallow water region can be converted back to overland flow regions.

2.2 Governing Equations

Surface runoff and flood waves in a watershed can be studied as *unsteady, open channel* flows in the context of *shallow water theory*. *Unsteady* flow means that the flow is highly variable with time. *Open channel* flow means that the flow is not entirely contained within rigid boundaries. The top surface of the flow is a free surface, denoting that the boundary is not in contact with a solid boundary and hence is freely deformable. *Shallow water theory* describes the hydraulics of flow where the vertical scale of the flow is much smaller than the horizontal scale. This theory applies to problems that are adequately described by the assumption that vertical accelerations are negligible [65]. The basic equations that govern shallow water theory, called the Saint Venant or shallow water equations, include balance laws for mass and momentum that are derived under the assumption of a hydrostatic pressure distribution. Simpler reduced models like diffusion wave model and kinematic wave model can be obtained from the basic equations by making additional assumptions. As described in Section 2.1, our modeling approach involves three different sets of equations which are described below.

2.2.1 One Dimensional Saint Venant Equations

The 1-D Saint Venant equations are widely used in engineering practices to model flow in rivers and open channels [65]. They consist of the following balance laws for two section-averaged quantities, viz. the wet cross-sectional area (A) and the mean cross-sectional volumetric discharge (Q). See Appendix A for the derivation of these equations.

$$\begin{aligned} \frac{\partial A}{\partial t} + \frac{\partial Q}{\partial s} &= q_L \\ \frac{\partial Q}{\partial t} + \frac{\partial}{\partial s}\left(\beta \frac{Q^2}{A} + gI_1\right) &= g(I_2 + AS_0 - AS_f) \end{aligned} \tag{2.1}$$

In the equations above, s represents the curvilinear coordinate along the center-line of the river or the channel; A, given in m^2, is the wet cross-sectional area; Q is the mean cross-sectional volumetric discharge and is given in m^3/s; q_L is the lateral inflow rate per unit length of channel, and is given in $\mathrm{m}^3/\mathrm{s}/\mathrm{m} = \mathrm{m}^2/\mathrm{s}$; g is the gravitational constant with units m^2/s; β is a dimensionless momentum correction factor that accounts for the fact that the velocity might not actually be uniform in the vertical direction; S_0 is the bed slope; S_f is the friction slope.

Natural channels and streams are often modeled with either rectangular or trapezoidal cross-sections. For channels with rectangular cross-sections, $A = WH$, where W is the width of the channel and H is the height of the water in the channel. For channels with trapezoidal cross-section, $A = W_b H + \frac{1}{2}H^2(m_1 + m_2)$, where H is the height of the water in the channel, W_b is the width of the channel at the bottom and m_1 and m_2 are the reciprocals of the

slopes of the two sides of the trapezoid. $Q = \bar{u}A$, where $\bar{u}$ is the mean flow velocity along the centerline and is given in m/s. $S_0 = \frac{\partial z}{\partial s}$, where z is the bathymetric depth measured from the geoid and is positive downwards. S_f is often approximated with Manning's formula: $S_f = \frac{n^2 Q|Q|}{A^2 H^{\frac{4}{3}}}$, where n is the Manning's roughness coefficient, given in $\mathrm{m}^{-\frac{1}{3}}\mathrm{s}$ [15]. The values of n have been tabulated from empirical studies. The momentum correction factor is given by: $\beta = \frac{1}{U^2 H} \int_A u^2 \, dA$, where A is the cross-section of the channel and U is the averaged velocity given by: $U = \frac{1}{A} \int_A u \, dA$. For any velocity distribution, β is greater than one. However, it is a common practice to set it equal to one [65].

I_1 accounts for hydrostatic pressure and I_2 accounts for wall pressure. These pressure terms are given by the following integral expressions:

$$\begin{aligned} I_1 &= \int_0^H (H - y) W(s, y) dy, \\ I_2 &= \int_0^H (H - y) \frac{\partial}{\partial s} \left(W(s, y) \right) dy. \end{aligned} \tag{2.2}$$

Note that in the expressions for I_1 and I_2 above, W is defined as the width of the channel at a particular vertical coordinate, y.

For channels with rectangular cross-sections, I_1 and I_2 can be simplified to the following expressions:

$$\begin{aligned} I_1 &= \frac{A^2}{2W}, \\ I_2 &= \frac{1}{2} \frac{A^2}{W^2} \frac{\partial W}{\partial s}. \end{aligned} \tag{2.3}$$

For channels with trapezoidal cross-sections, I_1 and I_2 can be simplified

to the following expressions:

$$
\begin{aligned}
I_1 &= H^2\frac{W_b}{2} + H^3\frac{m_1}{6} + H^3\frac{m_2}{6}, \\
I_2 &= \frac{H^2}{2}\frac{\partial W_b}{\partial s} + \frac{H^3}{6}\frac{\partial m_1}{\partial s} + \frac{H^3}{6}\frac{\partial m_2}{\partial s}.
\end{aligned}
\tag{2.4}
$$

where W_b is the width of the cross-section at the bottom of the channel, m_1 and m_2 are the slopes of the two sides of a channel cross-section.

For a prismatic channel, i.e. a channel whose width is constant along its length, $I_2 = 0$.

2.2.2 Two Dimensional Shallow Water Equations

The 2-D shallow water equations include balance laws for the water height (continuity equation) and the momentums in the x and the y directions (momentum equations). They can be derived by vertically integrating the 3-D Navier Stokes equations with the shallow water theory assumptions [65]. The continuity equation can be rewritten to express the conservation law for the water surface level as follows:

$$
\begin{aligned}
\frac{\partial \zeta}{\partial t} + \nabla \cdot \begin{pmatrix} Hu \\ Hv \end{pmatrix} &= q_L \\
\frac{\partial}{\partial t}(Hu) + \nabla \cdot \begin{pmatrix} Hu^2 + \frac{1}{2}g(H^2 - z^2) \\ Huv \end{pmatrix} &= g\zeta\frac{\partial z}{\partial x} - g\tau Hu + F_x \\
\frac{\partial}{\partial t}(Hv) + \nabla \cdot \begin{pmatrix} Huv \\ Hv^2 + \frac{1}{2}g(H^2 - z^2) \end{pmatrix} &= g\zeta\frac{\partial z}{\partial y} - g\tau Hv + F_y
\end{aligned}
\tag{2.5}
$$

In the equations above, ζ is the elevation of the free surface measured from the geoid and is positive upward; z, is the bathymetric depth measured from the geoid and is positive downwards; $H = \zeta + z$ is the total height of the

water column; g is the gravitational constant; u and v are the depth-averaged velocities in the x and y directions respectively; τ is the bottom friction factor, estimated using Manning's relation: $\tau = \left(\frac{n^2\sqrt{u^2+v^2}}{H^{4/3}}\right)$, where n is the Manning's roughness coefficient given in $\mathrm{m}^{-\frac{1}{3}}\mathrm{s}$. q_L is the lateral inflow rate given in m/s. In junctions, this term will be set equal to 0. It will be turned on in shallow water regions while simulating rainfall. In the case of rainfall, $q_L = R - I$, where $R = R(x, y, t)$ is the rainfall intensity and $I = I(x, y, t)$ is the infiltration rate, both of which are given in m/s. F_x and F_y denote additional terms that arise because of external forces like tidal potential forces, surface stresses such as wind or wave radiation stresses and Coriolis force if the simulation domain is large enough. Since these forces do not play a major role in driving the flow in junctions and floodplains, in those regions, we set this term equal to 0. In the coastal shelf, however, these terms play a very important role and some of these terms are the forces that drive storm surge. Thus, these terms cannot be neglected in shallow water regions near the coast.

2.2.3 Kinematic Wave Equation

For reasons described in Section 2.1, kinematic wave equation is widely used to model overland flow. Kinematic wave equation is a one-dimensional equation for modeling free surface flow. It is built on the assumption that there exists a balance between the gravitational and the frictional forces on the flow. Below, we present a quick derivation of the kinematic wave equation from the one-dimensional shallow water equations.

Overland flow can be described as channel flow in a wide rectangular channel. Substituting $A = WH$ in the continuity equation in (2.1) and dividing through by W, we obtain:

$$\frac{\partial H}{\partial t} + \frac{\partial (Hu)}{\partial s} = q_L \tag{2.6}$$

Note that in the equation above, s is the coordinate in the direction of the surface slope, and the lateral inflow rate q_L is given in m/s. Similarly to the 2-D shallow water equations, $q_L = R - I$, where $R = R(s,t)$ is the rainfall intensity and $I = I(s,t)$ is the infiltration rate, both of which are given in m/s. Now, assuming that there exists a balance between the gravitational force (S_0) and the frictional force (S_f) in the momentum equations of (2.1), we get:

$$S_f = S_0 \qquad \Rightarrow \qquad u = \frac{1}{n}\sqrt{S_0} H^{\frac{2}{3}} \tag{2.7}$$

Substituting (2.7) into (2.6), we obtain the kinematic wave equation:

$$\frac{\partial H}{\partial t} + \frac{\partial}{\partial s}\left(\frac{1}{n}\sqrt{S_0} H^{\frac{5}{3}}\right) = q_L \tag{2.8}$$

2.3 Discretization of the Governing Equations

For brevity of notation, we will denote the quantities associated with the 1-D Saint Venant equations, the 2-D shallow water equations and the kinematic wave equation as $(.)^{1D}$, $(.)^{2D}$ and $(.)^{Kin}$ respectively. All of the equations used in our model can be cast into the following divergence form:

$$\frac{\partial \mathbf{w}_i}{\partial t} + \nabla \cdot \mathbf{F}_i(\mathbf{w}) = \mathbf{S}_i(\mathbf{w}) \tag{2.9}$$

In the above equation, $i = 1$ for the kinematic wave model, $i = 1, 2$ for the 1-D Saint Venant equations and $i = 1, 2, 3$ for the 2-D shallow water equations. $\mathbf{w}_i$ is the i-th component of the vector $\mathbf{w}$ of conserved variables:

$$\mathbf{w}^{Kin} = H \qquad \mathbf{w}^{1D} = \begin{pmatrix} A \\ Q \end{pmatrix} \qquad \mathbf{w}^{2D} = \begin{pmatrix} \zeta \\ Hu \\ Hv \end{pmatrix}$$

$\mathbf{F}_i$ is the i-th row of the flux function matrix:

$$\mathbf{F}^{Kin} = \left[\frac{1}{n} \sqrt{S_0} H^{\frac{5}{3}} \right]$$

$$\mathbf{F}^{1D} = \begin{bmatrix} Q \\ \beta \frac{Q^2}{A} + g I_1 \end{bmatrix}$$

$$\mathbf{F}^{2D} = [\mathbf{f}_x, \mathbf{f}_y] = \begin{bmatrix} Hu & Hv \\ Hu^2 + \frac{1}{2} g (H^2 - z^2) & Huv \\ Huv & Hv^2 + \frac{1}{2} g (H^2 - z^2) \end{bmatrix}$$

$$\nabla \cdot \mathbf{F}_i^{Kin} = \nabla \cdot \mathbf{F}^{Kin} = \frac{\partial \mathbf{F}^{Kin}}{\partial s}$$

$$\nabla \cdot \mathbf{F}_i^{1D} = \frac{\partial \mathbf{F}_i^{1D}}{\partial s}$$

$$\nabla \cdot \mathbf{F}_i^{2D} = \frac{\partial \mathbf{f}_x}{\partial x} + \frac{\partial \mathbf{f}_y}{\partial y}$$

$\mathbf{S}_i$ is the i-th component of the vector $\mathbf{S}$ of source/sink terms:

$$\mathbf{S}^{Kin} = q_L \qquad \mathbf{S}^{1D} = \begin{pmatrix} q_L \\ g(I_2 + AS_0 - AS_f) \end{pmatrix} \qquad \mathbf{S}^{2D} = \begin{pmatrix} q_L \\ g\zeta \frac{\partial z}{\partial x} - g\tau Hu \\ g\zeta \frac{\partial z}{\partial y} - g\tau Hv \end{pmatrix}$$

The equation (2.9) represents a system of hyperbolic equations, and we will solve these equations using RKDG methods. Below, we present a brief discussion of a general RKDG scheme for the numerical solution of (2.9).

2.3.1 Discontinuous Galerkin Finite Element Method

Given a domain Ω with boundary $\partial\Omega$, let $\Omega_h = \bigcup_{e=0}^{N-1} \Omega_e$ be the finite element partition of Ω into a set of non-overlapping elements Ω_e. Let $\partial\Omega_e$ denote the boundary of Ω_e. Let $\mathbf{V^h} = \{v \in L^2(\Omega) : \ v|_{\Omega_e} \in \mathcal{P}^p(\Omega_e) \quad \forall\ \Omega_e \in \Omega_h\}$. Here, $\mathcal{P}^p$ denotes the space of polynomials of degree p defined on Ω_e. As such, the functions in $\mathbf{V^h}$ are allowed to be discontinuous across element edges. Now, multiplying (2.9) by $\psi \in \mathbf{V^h}$ and integrating over Ω_e, we get:

$$\left(\frac{\partial \mathbf{w}_i}{\partial t}, \psi\right)_{\Omega_e} + (\nabla \cdot \mathbf{F}, \psi)_{\Omega_e} = (\mathbf{S}_i, \psi)_{\Omega_e} \tag{2.10}$$

The notation $(\cdot,\cdot)_{\Omega_e}$ means the standard L^2 inner product over Ω_e for the 2-D case. For the 1-D case $(f,g)_{\Omega_e}$ represents the line integral $\int_{\mathbf{s}_e}^{\mathbf{s}_{e+1}} fg \ \mathrm{d}\mathbf{s}$. Integrating (2.10) by parts and rearranging the terms results in the following:

$$\left(\frac{\partial \mathbf{w}_i}{\partial t}, \psi\right)_{\Omega_e} = -\left\langle \hat{\mathbf{F}}_i(\mathbf{w}) \cdot \mathbf{n}, \psi \right\rangle_{\partial\Omega_e} + (\mathbf{F}_i(\mathbf{w}), \nabla\psi)_{\Omega_e} + (\mathbf{S}_i(\mathbf{w}), \psi)_{\Omega_e} \tag{2.11}$$

where $\hat{\mathbf{F}}$ represents a numerical flux that is defined such that it is continuous across element boundaries. $\mathbf{n} = \begin{pmatrix} n_x \\ n_y \end{pmatrix}$ represents the outward unit normal vector of $\partial\Omega_e$. The notation $< \cdot, \cdot >_{\partial\Omega_e}$ denotes the the boundary inner product. Note that for the 1-D case, $\left\langle \hat{\mathbf{F}}_i(\mathbf{w}) \cdot \mathbf{n}, \psi \right\rangle_{\partial\Omega_e} = \hat{\mathbf{F}}_i \psi|_{\mathbf{s_e}}^{\mathbf{s_{e+1}}}$.

Now, following the standard Galerkin method, we will approximate our unknowns $\mathbf{w}_i$ in space by $\mathbf{w^h}_i \in \mathbf{V^h}$. To ensure that our scheme is well-balanced, we will also approximate z by $z^h \in \mathbf{V^h}$. By substituting these approximations into (2.11), we obtain the following DG equations for each

element in the domain:

$$\left(\frac{\partial \mathbf{w}_i^{\mathbf{h}}}{\partial t}, \psi\right)_{\Omega_e} = -\left\langle \hat{\mathbf{F}}_i^{\mathbf{h}} \cdot \mathbf{n}, \psi \right\rangle_{\partial\Omega_e} + \left(\mathbf{F}_i^{\mathbf{h}}, \nabla\psi\right)_{\Omega_e} + \left(\mathbf{S}_i^{\mathbf{h}}(\mathbf{w}^{\mathbf{h}}), \psi\right)_{\Omega_e} \tag{2.12}$$

where $\mathbf{S}_i^{\mathbf{h}}$ indicates that the data terms that appear in the source terms are also approximated in $\mathbf{V}^{\mathbf{h}}$. $\mathbf{F}^{\mathbf{h}}$ and $\hat{\mathbf{F}}^{\mathbf{h}}$ denote the approximations of the flux $\mathbf{F}$ and the numerical flux $\hat{\mathbf{F}}$ respectively in $\mathbf{V}^{\mathbf{h}}$.

For completeness, we will briefly discuss the spatial discretization of the 1-D and the 2-D equations in the following section. Our discretization follows the approach described by Hesthaven and Warburton in [44] and the reader is encouraged to read [44] for additional details.

2.3.2 1-D Spatial Discretization

Notice that (2.12) is a scalar equation written separately for each conserved variable $\mathbf{w}_i$ for $i = 1, 2$ in the 1-D case and $i = 1, , 2, 3$ in the 2-D case. Therefore, in the discussion that follows, we discuss our DG scheme for the scalar balance law:

$$\frac{\partial w}{\partial t} + \frac{\partial F}{\partial x} = s \tag{2.13}$$

The DG scheme for this equation is:

$$\left(\frac{\partial w^h}{\partial t}, \psi\right)_{\Omega_e} = \left(F^h, \frac{\partial \psi}{\partial x}\right)_{\Omega_e} - \hat{F}^h \psi|_{\mathbf{s_e}}^{\mathbf{s_{e+1}}} + (s, \psi)_{\Omega_e} \tag{2.14}$$

Now, on an the element Ω_e, we express the local solution as a polynomial of order p as follows:

$$w^{h,e}(x,t) = \sum_{i=1}^{N_p} w^{h,e}(x_i^e, t) l_i^e(x) \tag{2.15}$$

where $N_p = p + 1$. $x_i^e \in \Omega_e$ are the grid points at which the polynomial $w^{h,e}(x,t)$ is interpolated using the associated Lagrange polynomials ($l_i^e(x)$). These polynomials form a set of basis functions for $\mathcal{P}^p(\Omega_e)$. Hence, we refer to these polynomials as nodal basis functions. The approximation of the global solution, $w^h(x,t)$, is then a piece-wise p-th order polynomial resulting from the direct sum of the N local approximations, i.e.:

$$w^h(x,t) = \bigoplus_{e=0}^{N-1} w^{h,e}(x,t) \tag{2.16}$$

We define the approximation of the flux F in V as follows:

$$F^{h,e}(x,t) = \mathcal{I}_p(F(w^{h,e}(x,t))) = \sum_{i=1}^{N_p} F(w^{h,e}(x_i^e,t))\, l_i^e(x) \tag{2.17}$$

where $\mathcal{I}_p$ denotes the p-th order interpolation operator. The points x_i are the interpolation points.

Substituting (2.15) and (2.17) into (2.14) and replacing the test function ψ with each of the basis functions l_i, we obtain the following:

$$\begin{aligned} &\mathcal{M}_e \frac{d}{dt}\tilde{\mathbf{w}}_e = \mathcal{S}_e^T \tilde{\mathbf{F}}_e + \mathcal{E}\tilde{\hat{\mathbf{F}}}_e + \mathcal{M}\tilde{\mathbf{s}}_e \\ &\Rightarrow \frac{d}{dt}\tilde{\mathbf{w}}_e = \mathcal{M}_e^{-1}\mathcal{S}_e^T \tilde{\mathbf{F}}_e + \mathcal{M}_e^{-1}\mathcal{E}\tilde{\hat{\mathbf{F}}}_e + \tilde{\mathbf{s}}_e \end{aligned} \tag{2.18}$$

In the equations above:

$$\mathcal{M}_e^{ij} = (l_i^e, l_j^e)_{\Omega_e}$$
$$\mathcal{S}_e^{ij} = \left(l_i^e, \frac{dl_j^e}{dx}\right)_{\Omega_e}$$
$$\mathcal{E} = [-\mathbf{e}_{N_p} \quad \mathbf{e}_1]$$
$$\tilde{\mathbf{w}}_e = [w^{h,e}(x_1^e, t) \cdots w^{h,e}(x_{Np}^e, t)]^T$$
$$\tilde{\mathbf{F}}_e = [F^{h,e}(x_1^e, t) \cdots F^{h,e}(x_{Np}^e, t)]^T$$
$$\hat{\tilde{\mathbf{F}}}_e = [\hat{F}^{h,e}(x_1^e, t) \cdots \hat{F}^{h,e}(x_{Np}^e, t)]^T$$

$\mathbf{e}_i$ is an N_p long zero vector with a value of 1 in position i. Notice that $\mathcal{M}_e$ and $\mathcal{S}_e$ are both N_p x N_p matrices and $\mathcal{E}$ is an N_p x 2 matrix.

Now let us discuss the 1-D nodal basis functions. To begin, consider the following affine mapping of the reference element $\Omega_r = [-1, 1]$ to the element $\Omega_e = [x_l^e, x_r^e]$:

$$x(r) = x_l^e + \frac{1+r}{2} h_e, \qquad h_e = x_r^e - x_l^e \tag{2.19}$$

where the reference variable $r \in \Omega_r$. Let $\psi_n(r)$ represent the orthonormal basis functions or the modal basis functions defined on Ω_r, given by:

$$\psi_n(r) = \tilde{P}_{n-1}(r) = \frac{P_{n-1}(r)}{\sqrt{\gamma_n - 1}}$$

where $P_n(r)$ are the classic Legendre polynomials of order n and

$$\gamma_n = \frac{2}{2n+1}$$

is the normalization. These basis functions can be computed through the following recursive relation:

$$r\tilde{P}_n(r) = a_n \tilde{P}_{n-1}(r) + a_{n+1}\tilde{P}_{n+1}(r)$$
$$a_n = \sqrt{\frac{n^2}{(2n+1)(2n-1)}}$$

with:

$$\psi_1(r) = \tilde{P}_0(r) = \frac{1}{\sqrt{2}}, \qquad \psi_2(r) = \tilde{P}_1(r) = \sqrt{\frac{3}{2}}r$$

Now, define:

$$\mathcal{V}_{ij} = \tilde{P}_{j-1}(r_j)$$

The matrix $\mathcal{V}$ is a generalized Vandermonde matrix. The points r_j are the Legendre–Gauss–Lobatto quadrature points and are defined as the N_p zeros of the polynomial

$$f(r) = (1 - r^2)\,\tilde{P}'_N(r)$$

Because of the uniqueness of interpolating polynomials, we can derive the following relationship between the nodal and the modal basis functions:

$$\mathcal{V}^T\,\mathbf{l}(r) = \tilde{\mathbf{P}}(r)$$

where $\mathbf{l}(r) = [l_1(r)\cdots l_{N_p}(r)]^T$ and $\tilde{\mathbf{P}}(r) = [P_1(r)\cdots P_{N_p}(r)]^T$. Furthermore, the mass matrix on the reference element, $M = M_{\Omega_r}$ can be calculated as:

$$M = (\mathcal{V}\mathcal{V}^T)^{-1} \tag{2.20}$$

and using the transformation given in (2.19), the mass matrix on the physical element is then calculated as:

$$M_e = \frac{h_e}{2}M \tag{2.21}$$

Define the differentiation matrix:

$$\mathcal{D}_r = \left.\frac{dl_j}{dr}\right|_{r_i}$$

It can be evaluated as:

$$\mathcal{D}_r = \mathcal{V}_r \mathcal{V}^{-1} \tag{2.22}$$

where, $\mathcal{V}_r^{ij} = \frac{d\tilde{P}_j}{dr}\Big|_{r_i}$. It an be shown that:

$$\mathcal{M}\mathcal{D}_r = \mathcal{S} \tag{2.23}$$

Note that $\mathcal{S}_{\Omega_e} = \mathcal{S}$.

Now, using (2.20), (2.21), (2.22) and (2.23), we can evaluate the right hand side of (2.18). The specific details of the numerical flux that we use in different sub-domains will be discussed in the chapter pertaining to that particular sub-domain.

2.3.3 2-D Spatial Discretization

Now consider a scalar balance law over a two dimensional domain written as:

$$\frac{\partial w}{\partial t} + \nabla \cdot \mathbf{F} = s \tag{2.24}$$

where, $\mathbf{F} = [F_x, F_y]$. The DG scheme for this equation is then:

$$\left(\frac{\partial w^h}{\partial t}, \psi\right)_{\Omega_e} = \left(\mathbf{F}^h, \nabla\psi\right)_{\Omega_e} - \left\langle \hat{\mathbf{F}}^h \cdot \mathbf{n}, \psi \right\rangle_{\partial\Omega_e} + (s, \psi)_{\Omega_e} \tag{2.25}$$

The elements Ω_e that discretize our domain Ω are now triangular. On an element Ω_e, we express the local solution as a polynomial of order p as follows:

$$w^{h,e}(\mathbf{x}, t) = \sum_{i=1}^{N_p} w^{h,e}(\mathbf{x}_i^e, t) l_i^e(\mathbf{x}) \tag{2.26}$$

where $N_p = \frac{(N+1)(N+2)}{2}$. $\mathbf{x}_i$ are the grid points at which the polynomial $w^{h,e}(\mathbf{x},t)$ is interpolated using the associated two-dimensional Lagrange polynomials $l_i^e(\mathbf{x})$. Similarly, we define the approximation of the flux $F \in V$ as follows:

$$\mathbf{F}^{h,e}(\mathbf{x},t) = \mathcal{I}_p(\mathbf{F}(w^{h,e}(\mathbf{x},t))) = \sum_{i=1}^{N_p} \mathbf{F}(w^{h,e}(\mathbf{x}_i^e,t))\, l_i^e(\mathbf{x}) \tag{2.27}$$

Now, substituting (2.26) and (2.27) into (2.25) and replacing the test function ψ with each of the basis functions l_i, we obtain the following system of equations:

$$\begin{aligned} &\mathcal{M}_e \frac{d}{dt}\tilde{\mathbf{w}}_e = \mathcal{S}_{x,e}^T \tilde{\mathbf{F}}_{x,e} + \mathcal{S}_{y,e}^T \tilde{\mathbf{F}}_{y,e} - \mathcal{E}_e \hat{\tilde{\mathbf{F}}}_{n,e} + \mathcal{M}_e \tilde{\mathbf{s}} \\ &\Rightarrow \frac{d}{dt}\tilde{\mathbf{w}}_e = \mathcal{M}_e^{-1}\mathcal{S}_{x,e}^T \tilde{\mathbf{F}}_{x,e} + \mathcal{M}_e^{-1}\mathcal{S}_{y,e}^T \tilde{\mathbf{F}}_{y,e} - \mathcal{M}_e^{-1}\mathcal{E}_e \hat{\tilde{\mathbf{F}}}_{n,e} + \tilde{\mathbf{s}} \end{aligned} \tag{2.28}$$

In the equation above:

$$\begin{aligned} \mathcal{M}_e^{ij} &= (l_i^e, l_j^e)_{\Omega_e} \\ \mathcal{S}_{x,e}^{ij} &= \left(l_i^e, \frac{\partial l_j^e}{\partial x}\right)_{\Omega_e} \\ \mathcal{S}_{y,e}^{ij} &= \left(l_i^e, \frac{\partial l_j^e}{\partial y}\right)_{\Omega_e} \\ \tilde{\mathbf{w}}_e &= [w^{h,e}(\mathbf{x}_1^e,t) \cdots w^{h,e}(\mathbf{x}_{Np}^e,t)]^T \\ \tilde{\mathbf{F}}_{x,e} &= [F_x^{h,e}(\mathbf{x}_1^e,t) \cdots F_x^{h,e}(\mathbf{x}_{Np}^e,t)]^T \\ \tilde{\mathbf{F}}_{y,e} &= [F_y^{h,e}(\mathbf{x}_1^e,t) \cdots F_y^{h,e}(\mathbf{x}_{Np}^e,t)]^T \\ \hat{\tilde{\mathbf{F}}}_{n,e} &= [\widehat{\mathbf{F}\cdot\mathbf{n}}^{h,e}(\mathbf{x}_1^e,t) \cdots \widehat{\mathbf{F}\cdot\mathbf{n}}^{h,e}(\mathbf{x}_{Np}^e,t)]^T \end{aligned}$$

Notice that $\mathcal{M}_e$, $\mathcal{S}_{x,e}$ and $\mathcal{S}_{y,e}$ are N_p x N_p matrices. The matrix $\mathcal{E}_E$ is a N_p x $3Nf_p$ matrix, where Nf_p represents the number of nodes on an edge of a triangle. For a polynomial approximation of order p, $Nf_p = p+1$. For edge

k of a triangular element, denoted e_k, define the Vandermonde matrix $\mathcal{V}^{1D}$ corresponding to the one-dimensional interpolation along the edge. Define also the edge mass matrix as:

$$\mathcal{M}_{e,e_k}^{ij} = \langle l_i, l_j \rangle_{e_k}$$

This matrix can be evaluated as:

$$\mathcal{M}_{e,e_k} = \mathcal{J}_{e,e_k}(\mathcal{V}^{1D}(\mathcal{V}^{1D})^T)^{-1}$$

where $\mathcal{J}_{e,e_k}$ is the ratio between the length of the face in Ω_e and in I respectively. The matrix $\mathcal{E}_e$ can be formed by inserting the elements of the edge mass matrices for each of the three edges of the triangle into the positions corresponding to the edge nodes of the element. We will discuss the two-dimensional basis functions and the interpolation points before describing how the matrices $\mathcal{M}_e$, $\mathcal{S}_x$ and $\mathcal{S}_y$ are formed.

The 2-D reference element is a straight-sided triangle, Ω_r, defined as:

$$\Omega_r = \{\mathbf{r} = (r, s) | (r, s) \geq -1; r + s \leq 0\}$$

The reference element can be mapped to any physical element Ω_e with vertices $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ through the map $\Psi(\mathbf{r})$ as such:

$$\mathbf{x} = \Psi(\mathbf{r}) = -\frac{r+s}{2}\mathbf{v}^1 + \frac{r+1}{2}\mathbf{v}^2 + \frac{s+1}{2}\mathbf{v}^3 \tag{2.29}$$

Just like in the 1-D case, we define the Vandermonde matrix $\mathcal{V}$ such that:

$$\mathcal{V}_{ij} = \psi_j(\mathbf{r}_i)$$

where $\mathbf{r}_i$ are the points in the reference triangle that map to the interpolation points ($\mathbf{x}_i$) in the physical element and ψ_j are the 2-D polynomial basis defined on Ω_r as follows:

$$\psi_m(\mathbf{r}) = \sqrt{2}P_i(a)P_j^{(2i+1,0)}(b)(1-b)^i$$

where

$$a = 2\frac{1+r}{1-s} - 1, \qquad b = s$$

and $P_n^{\alpha,\beta}$ is the n-th order Jacobi polynomial. On an equilateral triangle, if we define the interpolation points to be the equidistant points, the resulting Vandermonde matrix will have a large condition number, resulting in an unstable interpolation. However, we can compute the optimal grid points for interpolation by mapping these equidistant points to points that result in a well-conditioned Vandermonde matrix. This is inspired by the 1-D mapping

$$w(r) = \sum_{i=1}^{N_p}(r_i^{LGL} - r_i^e)l_i^e(r)$$

where r^{LGL} are the Legendre–Gauss–Lobatto points, r_i^e are the equidistant points in the interval $[0, \cdots, N]$ and l_i^e are the Lagrange polynomials associated with r_i^e. Let λ^k denote the barycentric coordinates such that any point $\mathbf{x}$ in a triangle with vertices $\mathbf{v}^1$, $\mathbf{v}^2$ and $\mathbf{v}^3$ can be expressed as:

$$\mathbf{x} = \lambda^2\mathbf{v}^1 + \lambda^3\mathbf{v}^2 + \lambda^1\mathbf{v}^3$$

Let $\mathcal{T}$ be an equilateral triangle with vertices $\mathbf{v}^1 = -\begin{pmatrix}1\\1/\sqrt{3}\end{pmatrix}$, $\mathbf{v}^2 = \begin{pmatrix}1\\-1/\sqrt{3}\end{pmatrix}$ and $\mathbf{v}^3 = \begin{pmatrix}0\\2/\sqrt{3}\end{pmatrix}$. The equidistant grid on this triangle can be defined using

barycentric coordinates as follows:

$$(\lambda^1, \lambda^3) = \left(\frac{i}{p}, \frac{j}{p}\right), \quad \lambda^2 = 1 - \lambda^1 - \lambda^3 \qquad \text{for } (i,j) \geq 0,\ i+j \leq p \quad (2.30)$$

Define the following warping functions:

$$\mathbf{w}^1(\lambda^1, \lambda^2, \lambda^3) = \tilde{w}(\lambda^3 - \lambda^2)\begin{pmatrix}1\\0\end{pmatrix}$$
$$\mathbf{w}^2(\lambda^1, \lambda^2, \lambda^3) = \tilde{w}(\lambda^1 - \lambda^3)\frac{1}{2}\begin{pmatrix}-1\\\sqrt{3}\end{pmatrix}$$
$$\mathbf{w}^3(\lambda^1, \lambda^2, \lambda^3) = \tilde{w}(\lambda^2 - \lambda^1)\frac{1}{2}\begin{pmatrix}-1\\-\sqrt{3}\end{pmatrix}$$

where $\tilde{w}(r) = \frac{w(r)}{1-r^2}$. Now, define the following blending functions:

$$b^1(\lambda^1, \lambda^2, \lambda^3) = 4\lambda^3\lambda^2$$
$$b^1(\lambda^1, \lambda^2, \lambda^3) = 4\lambda^3\lambda^1$$
$$b^1(\lambda^1, \lambda^2, \lambda^3) = 4\lambda^2\lambda^1$$

The equidistant points can now be transformed into better behaved points by the following transformation:

$$\mathbf{x} = \mathbf{x} + b^1\mathbf{w}^1 + b^2\mathbf{w}^2 + b^3\mathbf{w}^3 \quad (2.31)$$

where $\mathbf{x} = \begin{pmatrix}x\\y\end{pmatrix}$. Since these grid points are defined on the equilateral triangle, we need to map these points to the points (r, s) on the reference element I.

This can be done with the following map:

$$\begin{aligned}
\lambda^1 &= \frac{\sqrt{3}y + 1}{3} \\
\lambda^2 &= \frac{-3x - \sqrt{3}y + 2}{6} \\
\lambda^3 &= \frac{3x - \sqrt{3}y + 2}{6} \\
r &= -\lambda^2 + \lambda^3 - \lambda^1 \\
s &= -\lambda^2 - \lambda^3 + \lambda^1
\end{aligned} \tag{2.32}$$

We now have a set of orthonormal basis functions and the optimal grid points for interpolation. Let us proceed to form the desired matrices:

$$\begin{aligned}
\mathcal{M}_e &= \mathcal{J}_e \mathcal{M} = \mathcal{J}_e (\mathcal{V}\mathcal{V}^T)^{-1} \\
\mathcal{S}_{x,e} &= \mathcal{J}_e (r_x \mathcal{S}_r + s_x \mathcal{S}_s) \\
\mathcal{S}_{y,e} &= \mathcal{J}_e (r_y \mathcal{S}_r + s_y \mathcal{S}_s)
\end{aligned} \tag{2.33}$$

where $\mathcal{J}_e$ is the determinant of the Jacobian matrix of the transformation (2.29) and:

$$\begin{aligned}
r_x &= \frac{\partial r}{\partial x} & r_y &= \frac{\partial r}{\partial y} \\
s_x &= \frac{\partial s}{\partial x} & s_y &= \frac{\partial s}{\partial y} \\
\mathcal{S}_r &= \left(l_i, \frac{\partial l_j}{\partial r} \right)_{\Omega_r} & \mathcal{S}_s &= \left(l_i, \frac{\partial l_j}{\partial s} \right)_{\Omega_r}
\end{aligned}$$

Now, the matrices $\mathcal{S}_r$ and $\mathcal{S}_r$ can be computed as follows:

$$\mathcal{S}_r = \mathcal{M}^{-1} \mathcal{D}_r \qquad \mathcal{S}_s = \mathcal{M}^{-1} \mathcal{D}_s \tag{2.34}$$

where $\mathcal{D}_r$ and $\mathcal{D}_s$ are the differentiation matrices defined as follows:

$$\mathcal{D}_r^{ij} = \left. \frac{\partial l_j}{\partial r} \right|_{\mathbf{r}_i} \qquad \mathcal{D}_s^{ij} = \left. \frac{\partial l_j}{\partial s} \right|_{\mathbf{r}_i}$$

where $\mathbf{r}_i$ are the interpolation points. Now, define $\mathcal{V}_r^{ij} = \left.\frac{\partial \psi_j}{\partial r}\right|_{\mathbf{r}_i}$ and $\mathcal{V}_s^{ij} = \left.\frac{\partial \psi_j}{\partial s}\right|_{\mathbf{r}_i}$. Then, we can calculate the differentiation matrices as follows:

$$\mathcal{D}_r = \mathcal{V}_r \mathcal{V}^{-1} \qquad \mathcal{D}_s = \mathcal{V}_s \mathcal{V}^{-1} \tag{2.35}$$

Now, using (2.33), (2.34) and (2.35), we can evaluate the right hand side of (2.28).

This completes the discussion of the spatial discretization of the 2-D equations. In the next section, we will discuss the time discretization of the 1-D as well as the 2-D equations.

2.3.4 Time Discretization

We discretize the set of ODEs given in (2.18) and (2.28) using a strong stability preserving (SSP) explicit m-stage Runge–Kutta time stepping method. For $m \leq 4$, these methods are m-th order accurate and they can be written in the Shu–Osher representation [86] as follows:

$$\tilde{\mathbf{w}}^{(j)} = \sum_{k=0}^{j-1} \left[\alpha_{jk} \tilde{\mathbf{w}}^{(k)} + \beta_{jk} \Delta t L(\tilde{\mathbf{w}}^{(k)})\right] \tag{2.36}$$

where, $j = 1 \cdots m$, $\tilde{\mathbf{w}}^{(0)} = \tilde{\mathbf{w}}^n$, $\tilde{\mathbf{w}}^m = \tilde{\mathbf{w}}^{n+1}$, and $\tilde{\mathbf{w}}^n$ is the value of $\tilde{\mathbf{w}}$ at the n-th time step. The coefficients α_{jk} are required to satisfy the constraint: $\sum_{k=0}^{j-1} \alpha_{jk} = 1$. The coefficients $\beta_{jk} = c_{jk} - \sum_{l=k+1}^{j-1} c_{lk}\alpha_{jl}$, where $\sum_{k=0}^{m-1} c_{mk} = 1$ for consistency (see [14]). In order to render our scheme stable, we impose the following additional constraints on the coefficients:

i. $\alpha_{jk} \geq 0$ and $\beta_{jk} \geq 0$

ii. $\alpha_{jk} = 0$ only if $\beta_{jk} = 0$

The reader is referred to [86] for additional details. We use the second order Runge–Kutta method in our work. As with any other explicit scheme, the time steps are restricted by a CFL condition. The CFL number required for stability for an RKDG scheme is given by $\frac{1}{2p+1}$, where p is the order of the polynomial approximation used in space (see [18]).

Chapter 3

Flow in a Network of Channels[1]

In this chapter, we begin by describing how a network of channels is decomposed into channel segments and junctions. Then, we provide details of the numerical flux that we use in our RKDG scheme for solving the 1-D Saint Venant equations and the 2-D shallow water equations. We explain how the channels are coupled with junctions. Then, we present a theoretical analysis of the mass conservation property of the scheme and present a stability analysis of the coupling using slightly simplified equations. We conclude the chapter by presenting some numerical results to validate and verify our model. Much of the work presented in this chapter is published in [70].

3.1 Decomposition of Channel Network

Networks of streams and rivers exist naturally in a natural watershed. We model these networks as channel networks comprising of multiple channels that either meet at a junction or splinter off from a junction as shown in

[1]P. Neupane and C. Dawson. A discontinuous Galerkin method for modeling flow in networks of channels. *Advances in Water Resources*, 79:61-79, 2015. The author of this dissertation was responsible for the technical details of the model and producing all the results included in the paper.

Figure 3.1.

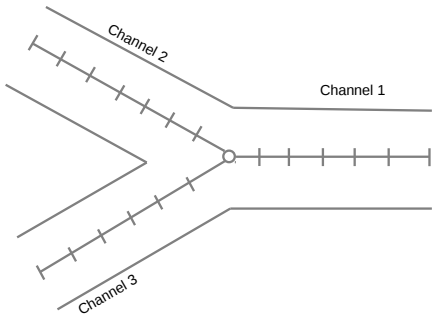


Figure 3.1: A schematic of a simple channel network comprising three channels and a junction.

We decompose the network of channels shown in Figure 3.1 into three 1-D channels that are connected to each other via a 2-D junction as shown in Figure 3.2. We require a channel to either start or end at a junction. No channel is allowed to have a junction somewhere along its length; if a channel meets another channel at a junction, the section that emerges after the junction is treated as a separate channel. The junction is created so that the centerline of the channel is perpendicular to the edges of the junction that are connected to the channels. This allows for a conservative coupling of the 1-D channels with the 2-D junctions. Once the domain is thus decomposed, we discretize channel segments by connecting the nodes on the centerline of the channel with straight lines, i.e. if $\mathbf{s_e} = \begin{pmatrix} x_e \\ y_e \end{pmatrix}$ and $\mathbf{s_{e+1}} = \begin{pmatrix} x_{e+1} \\ y_{e+1} \end{pmatrix}$ are the two vertices on the centerline of a channel segment, then we represent the element flanked by these two vertices by $\Omega_e = \{x \mid x = (1-t)\,\mathbf{s_e} + t\,\mathbf{s_{e+1}} \text{ for } t \in [0,1]\}$. We

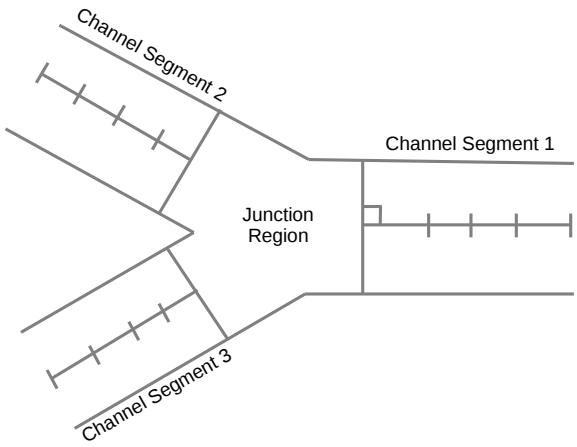


Figure 3.2: A schematic of a simple channel network comprised of three channels that are connected to each other via a junction

discretize the junction using an unstructured mesh with triangular elements. Then, we solve the 1-D Saint Venant equations given in (2.1) in the channels and the 2-D shallow water equations given in (2.5) in the junctions using the RKDG scheme described in Section (2.3). In the next section, we discuss the numerical fluxes that we use in our scheme.

3.2 Numerical Flux

Numerous types of fluxes have been developed in the finite volume community for conservation laws (See [62]). The numerical fluxes that result in a stable numerical scheme take advantage of the fact that the system (2.9) can be rewritten as a hyperbolic system, and that the domain of dependence of these systems are finite. We use Roe's flux and the local Lax-Friedrichs flux

in our work (See [77]).

Roe's flux is an approximate Riemann solver that resolves shocks accurately. Instead of solving the nonlinear set of equations (2.9), Roe's solver solves a linear system that approximates the original conservation law. The approximate linear system that is solved is the following:

$$\frac{\partial \mathbf{w}_i}{\partial t} + \mathbf{J}(\hat{\mathbf{w}})\mathbf{w}_i = \mathbf{S}_i(\mathbf{w}) \tag{3.1}$$

where $\mathbf{J}(\hat{\mathbf{w}})$ is the Jacobian matrix $(\frac{\partial \mathbf{F}_i}{\partial \mathbf{w}})$ evaluated at the Roe-average $\hat{\mathbf{w}}$.

Define $c \equiv \sqrt{gH}$. Define $\bar{u} \equiv \frac{Q}{A}$. Then the Jacobian matrices for the 1-D and the 2-D equations are given by:

$$\mathbf{J}^{1D} = \partial_{\mathbf{w}}\mathbf{f} = \begin{pmatrix} 0 & 1 \\ c^2 - \bar{u}^2 & 2\bar{u} \end{pmatrix} \tag{3.2}$$

$$\mathbf{J}_x^{2D} = \frac{\partial \mathbf{f}_x}{\partial \mathbf{w}} = \begin{pmatrix} 0 & 1 & 0 \\ c^2 - u^2 & 2u & 0 \\ -uv & v & u \end{pmatrix} \quad \mathbf{J}_y^{2D} = \frac{\partial \mathbf{f}_y}{\partial \mathbf{w}} = \begin{pmatrix} 0 & 0 & 1 \\ -uv & v & u \\ c^2 - v^2 & 0 & 2v \end{pmatrix} \tag{3.3}$$

The normal Jacobian matrices are defined as:

$$\mathbf{J}_n^{1D} = \mathbf{J}^{1D} \tag{3.4}$$

$$\mathbf{J}_n^{2D} = \mathbf{J}_x^{2D} n_x + \mathbf{J}_y^{2D} n_y = \begin{pmatrix} 0 & n_x & n_y \\ c^2 n_x - u^2 n_x - uvn_y & 2un_x + vn_y & un_y \\ -uvn_x + c^2 n_y - v^2 n_y & vn_x & un_x + 2vn_y \end{pmatrix} \tag{3.5}$$

These normal Jacobian matrices have the following real eigenvalues:

$$\lambda_1^{1D} = \bar{u} + c \qquad \lambda_2^{1D} = \bar{u} - c \tag{3.6}$$

$$\lambda_1^{2D} = un_x + vn_y - c \qquad \lambda_2^{2D} = un_x + vn_y \qquad \lambda_3^{2D} = un_x + vn_y + c \tag{3.7}$$

The corresponding eigenvectors are:

$$\mathbf{r}_1^{1D} = \begin{pmatrix} 1 \\ \bar{u} + c \end{pmatrix} \qquad \mathbf{r}_2^{1D} = \begin{pmatrix} 1 \\ \bar{u} - c \end{pmatrix} \tag{3.8}$$

$$\mathbf{r}_1^{2D} = \begin{pmatrix} 1 \\ u - cn_x \\ v - cn_y \end{pmatrix} \qquad \mathbf{r}_2^{2D} = \begin{pmatrix} 0 \\ -n_y \\ n_x \end{pmatrix} \qquad \mathbf{r}_3^{2D} = \begin{pmatrix} 1 \\ u + cn_x \\ v + cn_y \end{pmatrix} \tag{3.9}$$

Next, we define the following two matrices of eigenvalues and eigenvectors:

$$|\Lambda^{1D}| = \begin{pmatrix} |\lambda_1^{1D}| & 0 \\ 0 & |\lambda_2^{1D}| \end{pmatrix} \qquad \mathbf{R}^{1D} = \left(\mathbf{r}_1^{1D}, \mathbf{r}_2^{1D}\right) \tag{3.10}$$

$$|\Lambda^{2D}| = \begin{pmatrix} |\lambda_1^{2D}| & 0 & 0 \\ 0 & |\lambda_2^{2D}| & 0 \\ 0 & 0 & |\lambda_3^{2D}| \end{pmatrix} \qquad \mathbf{R}^{2D} = \left(\mathbf{r}_1^{2D}, \mathbf{r}_2^{2D}, \mathbf{r}_3^{2D}\right) \tag{3.11}$$

We are now ready to introduce Roe's flux, which is given by the expression below:

$$\hat{F}_n = \frac{1}{2}\left(F_n(\mathbf{w}^{\mathbf{h}})^{in} + F_n(\mathbf{w}^{\mathbf{h}})^{ex}\right) + \frac{1}{2}\hat{\mathbf{R}}|\hat{\Lambda}|\hat{\mathbf{R}}^{-1}\left((\mathbf{w}^{\mathbf{h}})^{in} - (\mathbf{w}^{\mathbf{h}})^{ex}\right) \tag{3.12}$$

where F_n is the normal flux whose i-th component is given by:

$$F_n^i = \mathbf{F}_i \cdot \mathbf{n}$$

and "in" represents the interior state and "ex" represents the exterior state.

Note that in the 1-D case, $\mathbf{F}_n = \mathbf{F}$. $\hat{\mathbf{R}}$ and $|\hat{\Lambda}|$ are the matrices defined in (3.10) and (3.11) evaluated at the Roe averages as detailed in [36]. The Roe averages are given by:

$$\hat{u}^{1D} \equiv \left(\frac{Q^{in}}{\sqrt{A^{in}}} + \frac{Q^{ex}}{\sqrt{A^{ex}}}\right)\left(\frac{1}{\sqrt{A^{in}} + \sqrt{A^{ex}}}\right) \qquad \hat{c}^{1D} \equiv \sqrt{\frac{g}{2}\left(\frac{A^{in} + A^{ex}}{W}\right)}$$

$$\hat{u}^{2D} \equiv \frac{u^{in}\sqrt{H^{in}} + u^{ex}\sqrt{H^{ex}}}{\sqrt{H^{in}} + \sqrt{H^{ex}}} \qquad \hat{v}^{2D} \equiv \frac{v^{in}\sqrt{H^{in}} + v^{ex}\sqrt{H^{ex}}}{\sqrt{H^{in}} + \sqrt{H^{ex}}} \qquad \hat{c}^{2D} \equiv \sqrt{\frac{g}{2}(H^{in} + H^{ex})}$$

Consequently, the flux given in (3.12) is continuous across element boundaries.

The local Lax-Friedrichs flux is a simpler flux given by the following expression:

$$\hat{F}_n = \frac{1}{2}\left(F_n(\mathbf{w}^{\mathbf{h}})^{in} + F_n(\mathbf{w}^{\mathbf{h}})^{ex}\right) + \frac{\lambda}{2}\left((\mathbf{w}^{\mathbf{h}})^{in} - (\mathbf{w}^{\mathbf{h}})^{ex}\right) \tag{3.13}$$

where $\lambda = \max_i \left(\max(|\lambda_i^{in}|, |\lambda_i^{ex}|)\right)$ for $i = 1,\ 2$ in the 1-D case and $i = 1,\ 2,\ 3$ in the 2-D case. Note that λ is positive and is single-valued at element boundaries.

Now that we have described how the equations governing flow in channels and junctions are solved separately, we are ready to discuss how these two sub-domains are coupled with each other to model flow in the entire network of channels.

3.3 Coupling of Channels with Junctions

The channels are coupled to each other by prescribing appropriate boundary conditions on the junctions and on the channels that meet at the junction. These boundary conditions are the interface conditions that couple the channels sub-domain with the junction sub-domain. These conditions have to account for the difference in the dimensions of the channel segments and the junction. The boundary conditions are prescribed as exterior states at the node of the channel connected to a junction and at the nodes lying on the edges of the junction that are connected to channels in the DG treatment at every time step. To keep the notation from getting cluttered, let $(\cdot)^{1D}$ denote

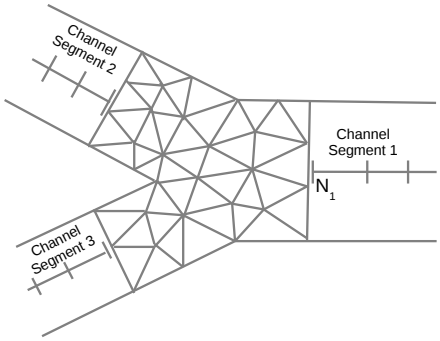


Figure 3.3: Junction region is meshed using triangular elements. N_1 represents the interface node of channel 1

.

the quantity $(\cdot)$ computed in the channel segments using the 1-D St. Venant equations and $(\cdot)^{2D}$ denote the quantity $(\cdot)$ computed in the junction using the 2-D shallow water equations.

Boundary conditions are prescribed on all of the outer edges of the 2-D junction. On the edges that are not connected to a channel, we prescribe a no flow boundary condition by specifying the following exterior conditions at the nodes that lie on those edges:

$$\zeta^{ex} = \zeta^{in} \tag{3.14}$$

$$Q_n^{2D,ex} = -Q_n^{2D,in} \tag{3.15}$$

$$Q_t^{2D,ex} = Q_t^{2D,in} \tag{3.16}$$

where Q_n^{2D} and Q_t^{2D} are the normal and tangential components of the flux at the respective nodes, i.e. they are the normal and tangential components of the vector $\mathbf{Q}^{2D} = \begin{pmatrix} Hu \\ Hv \end{pmatrix}$.

Recall that $\mathbf{Q}^{2D}$ is the flow rate measured in m^2/s and Q^{1D} is the

volumetric discharge measured in m^3/s. Therefore, while coupling the channels to the junctions, we need to make sure that the boundary conditions are dimensionally consistent. To see how this is done, let us direct our attention to the edges of the junction connected to channel segment 1 in Figure 3.3. Assume there are N such edges. Let each edge be denoted by e_i, where $i = 1 \cdots N$. Recall that on any triangular element of the junction, $\mathbf{Q}^{2D} = \sum_{i=1}^{N_p} \tilde{\mathbf{Q}}^{2D} \, l_i^{2D}$. The normal component of the flux on e_i is then given by:

$$Q_n^{2D} = \mathbf{Q}^{2D} \cdot \mathbf{n}$$

where $\mathbf{n}$ is the outward unit normal vector of edge e_i and n_x and n_y represent its $x-$ and $y-$ components. The volumetric discharge through this edge, denoted $Q_{vol,i}^{2D}$, can be calculated by the following line integral:

$$Q_{vol,i}^{2D} = \int_{e_i} Q_n^{2D} \, \mathrm{d}\mathbf{s}$$

This integrand can be computed as follows:

$$Q_{vol,i}^{2D} = \mathbf{1}^T \mathcal{M}_{e_i} \tilde{\mathbf{Q}}_{\mathbf{n},e_i}^{2D}$$

where $\mathbf{1}$ is a Nf_p dimensional vector of ones, $\mathcal{M}_{e_i}$ is the Nf_p x Nf_p edge mass matrix as described in Section 2.3.3, and $\tilde{\mathbf{Q}}_{\mathbf{n},e_i}^{2D}$ is the Nf_p vector containing the values of the interior states of the normal flux at the Nf_p nodes that lie on edge e_i.

The total volumetric discharge at the interface with channel segment 1 is then given by $\sum_{i=1}^{N} Q_{vol,i}^{2D} \equiv Q_{I,TV}^{2D}$. Thus, we prescribe the following boundary

condition on the interface node of channel segment 1:

$$Q^{1D,ex} n_{channel} = -Q^{2D,in}_{I,TV} \tag{3.17a}$$

where, $Q^{1D,ex}$ denotes the exterior state of the volumetric discharge at the interface node of the channel segment and $n_{channel}$ denotes the outward unit normal vector of the channel segment at the node. Note that $n_{channel}$ will be equal to -1 at the first node of a channel element and 1 at the last node of a channel element. The negative sign in (3.17a) indicates that the outward normal vectors corresponding to a channel segment and the junction connected to it have opposite signs.

We prescribe the following boundary condition on each node lying on the junction edge e_i:

$$Q^{2D,ex}_{n,i} = -\frac{Q^{1D,in}\, n_{channel}}{\sum_{i=1}^{N} |e_i|} = -\frac{Q^{1D,in}\, n_{channel}}{W_I} \tag{3.17b}$$

where $W_I = \sum_{i=1}^{N} |e_i|$ is the width of the channel segment at the interface.

Now, let us direct our attention to the interface conditions for the water height. Let A^{1D} be the area of the wet cross-section at the interface node of channel segment 1. Let $\tilde{\mathbf{H}}_i^{2D,in}$ denote the Nf_p vector containing the values of the interior states of water height at the Nf_p nodes on e_i. The average value of the water height on e_i can be calculated as:

$$H^{2D,in}_{avg,i} = \frac{1}{|e_i|} \int_{e_i} H_i^{2D,in}\, \mathbf{dx} = \frac{1}{|e_i|} \mathbf{1}^T \mathcal{M}_{e_i} \tilde{\mathbf{H}}_i^{2D,in}$$

where $|e_i|$ is the length of edge e_i.

Since the cross-section of channel segment 1 at the interface is parallel to the edge, the wet cross-sectional area at the edge is given by: $|e_i|H^{2D,in}_{avg,i}$. Thus, we prescribe the following boundary condition on the interface node of channel segment 1:

$$A^{1D,ex} = \sum_{i=1}^{N} |e_i| H^{2D,in}_{avg,i} \tag{3.17c}$$

Finally, we prescribe the following boundary condition on each of the Nf_p nodes that lie on junction edge e_i:

$$H^{2D,ex}_i = \frac{A^{1D,in}}{W_I} \tag{3.17d}$$

The four equations (3.17a) – (3.17d) collectively form the coupling conditions for the channel and the junction sub-domains. Notice that these interface conditions make $Q^{2D,ex}_n$ and $H^{2D,ex}$ constant on the interface edges.

We note that with these fluxes, the principle of conservation of mass is preserved throughout the channel network. We will prove this analytically in the next section.

3.4 Mass Conservation Property of Coupling

Here, we show that the conditions used to couple the 1-D channels with the 2-D junctions are conservative. For simplicity, we look at the case of two channels that are connected to each other via a junction in the middle as shown in Figure 3.4. Let $C1$ and $C2$ represent two channels that are connected to

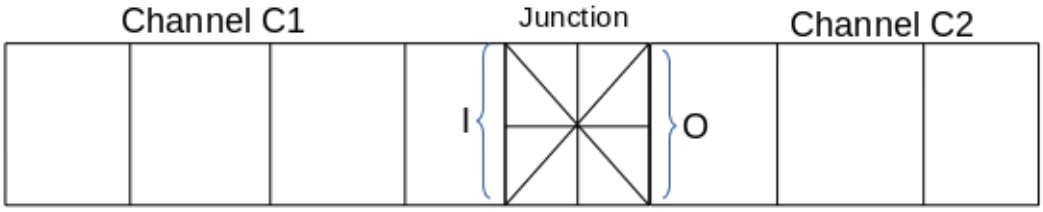


Figure 3.4: Schematic showing two channels that are connected by a junction in the middle

each other via a junction. Let I denote the exterior edge of the junction that is connected to $C1$ and let O denote the exterior edge of the junction that is connected to $C2$. Without loss of generality, assume that $C1$ is the inflow channel and $C2$ is the outflow channel. For the purposes of this analysis, we use the local Lax-Friedrichs flux as the numerical flux in the 1-D as well as the 2-D equations. To test for mass conservation, we enforce a no flow boundary condition on the inflow edge of $C1$ and on the outflow edge of $C2$.

Now, consider the weak formulation of the continuity equation for the channels:

$$\forall \psi \in V, \qquad \left\langle \frac{\partial A}{\partial t}, \psi \right\rangle_{s_i} = \hat{Q}\psi|_{\mathbf{s}_{i-\frac{1}{2}}} - \hat{Q}\psi|_{\mathbf{s}_{i+\frac{1}{2}}} + \left\langle Q, \frac{\partial \psi}{\partial x} \right\rangle_{s_i} \tag{3.18}$$

where s_i is a channel element that extends from vertex $\mathbf{s}_{i-\frac{1}{2}}$ to vertex $\mathbf{s}_{i+\frac{1}{2}}$, for $i = 1 \ldots n$, where n is the number of elements used to discretize the channel. Where boundary conditions are prescribed, we replace the numerical flux $\hat{Q}$ by the exact boundary condition, denoted Q_B.

Now, we replace ψ by the constant polynomial $1 \in V$ in (3.18) to

obtain:

$$\int_{s_i} \frac{\partial A}{\partial t}\ \mathbf{ds} = \hat{Q}|_{s-\frac{1}{2}} - \hat{Q}|_{s+\frac{1}{2}} \tag{3.19}$$

Summing up the above equation over all elements in the 1-D channel, recalling that the numerical flux, $\hat{Q}$, is continuous at element interfaces, and using our no-flow boundary conditions, we obtain the following for channel $C1$:

$$\begin{aligned} \sum_{i=1}^{n} \int_{s_i} \frac{\partial A}{\partial t}\ \mathbf{ds} &= Q_B|_{\mathbf{s}_{\frac{1}{2}}} - \hat{Q}|_{\mathbf{s}_{n+\frac{1}{2}}} \\ &= -\hat{Q}|_{\mathbf{s}_{n+\frac{1}{2}}} \end{aligned} \tag{3.20}$$

Similarly, for channel $C2$, we obtain:

$$\begin{aligned} \sum_{i=n+1}^{n+m} \int_{s_i} \frac{\partial A}{\partial t}\ \mathbf{ds} &= \hat{Q}|_{\mathbf{s}_{n+\frac{3}{2}}} - Q_B|_{\mathbf{s}_{n+m+\frac{1}{2}}} \\ &= \hat{Q}|_{\mathbf{s}_{n+\frac{3}{2}}} \end{aligned} \tag{3.21}$$

where s_i is an element of $C2$ that extends from vertex $\mathbf{s}_{i-\frac{1}{2}}$ to vertex $\mathbf{s}_{i+\frac{1}{2}}$, for $i = n+\frac{1}{2} \ldots n+m+\frac{1}{2}$, where m is the number of elements used to discretize $C2$.

Combining (3.20) and (3.21), we get the following expression for the total water mass (M_c) in the two channels:

$$\frac{d}{dt} M_c = -\hat{Q}|_{\mathbf{s}_{n+\frac{1}{2}}} + \hat{Q}|_{\mathbf{s}_{n+\frac{3}{2}}} \tag{3.22}$$

where $M_e \equiv \sum_{i=1}^{n+m} \int_{s_i} A\ \mathbf{ds}$.

The Lax-Friedrichs flux for (3.18) is given by:

$$\hat{Q} = \frac{1}{2}(Q^{in} + Q^{ex}) + \frac{\lambda n_{channel}}{2}(A^{in} - A^{ex}) \tag{3.23}$$

Substituting (3.23) into (3.22), we obtain:

$$\begin{aligned}\frac{d}{dt}M_c = &-\frac{1}{2}(Q^{in}_{\mathbf{s}_{n+\frac{1}{2}}} + Q^{ex}_{\mathbf{s}_{n+\frac{1}{2}}}) - \frac{\lambda^{1D}_{\mathbf{s}_{n+\frac{1}{2}}}}{2}(A^{in}_{\mathbf{s}_{n+\frac{1}{2}}} - A^{ex}_{\mathbf{s}_{n+\frac{1}{2}}}) + \frac{1}{2}(Q^{in}_{\mathbf{s}_{n+\frac{3}{2}}} + Q^{ex}_{\mathbf{s}_{n+\frac{3}{2}}}) \\ &+ \frac{\lambda^{1D}_{\mathbf{s}_{n+\frac{1}{2}}}}{2}(A^{ex}_{\mathbf{s}_{n+\frac{3}{2}}} - A^{in}_{\mathbf{s}_{n+\frac{3}{2}}})\end{aligned} \tag{3.24}$$

Substituting the interface conditions (3.17a) – (3.17d) into (3.24), we obtain:

$$\begin{aligned}\frac{d}{dt}M_c = &-\frac{1}{2}\left(Q^{1D,in}_{\mathbf{s}_{n+\frac{1}{2}}} - Q^{2D,in}_{I,TV}\right) - \frac{\lambda^I}{2}\left(A^{1D,in}_{\mathbf{s}_{n+\frac{1}{2}}} - \sum_i |e_i| H^{2D,in}_{avg,i}\right) \\ &+ \frac{1}{2}\left(Q^{1D,in}_{\mathbf{s}_{n+\frac{3}{2}}} + Q^{2D,in}_{O,TV}\right) + \frac{\lambda^O}{2}\left(-A^{1D,in}_{\mathbf{s}_{n+\frac{3}{2}}} + \sum_o |e_o| H^{2D,in}_{avg,o}\right)\end{aligned} \tag{3.25}$$

where λ^I and λ^O denote the maximum eigenvalues at the channel-junction interfaces I and O respectively. These quantities will be defined after the numerical flux for the 2-D equations is introduced.

Now, consider the weak formulation of the continuity equation for an element Ω_a of a junction:

$$\forall \psi \in V, \qquad \left(\frac{\partial \zeta}{\partial t}, \psi\right)_{\Omega_a} = -\left\langle \hat{\mathbf{Q}}^{2D} \cdot \mathbf{n}, \psi \right\rangle_{\partial\Omega_a} + \left(\mathbf{Q}^{2D}, \frac{\partial \psi}{\partial \mathbf{x}}\right)$$

where $\mathbf{n}$ is the outward unit vector normal to each edge. Equivalently,

$$\forall \psi \in V, \qquad \left(\frac{\partial H^{2D}}{\partial t}, \psi\right)_{\Omega_a} = -\left\langle \hat{\mathbf{Q}}^{2D} \cdot \mathbf{n}, \psi \right\rangle_{\partial\Omega_a} + \left(\mathbf{Q}^{2D}, \frac{\partial \psi}{\partial \mathbf{x}}\right) \tag{3.26}$$

Replacing ψ by the constant polynomial $1 \in V$ in (3.26), we obtain:

$$\int_{\Omega_a} \frac{\partial H}{\partial t}^{2D} \, \mathbf{dx} = -\int_{\partial\Omega_a} \hat{\mathbf{Q}}^{2D} \cdot \mathbf{n} \, \mathrm{ds} \tag{3.27}$$

Let E be an index that goes over the edges that lie on the exterior boundary of the domain (i.e. on $\partial\Omega$). Let e_i be an index that goes over all the small element edges of the junction that make up the big edge I, i.e. $I = \sum_i e_i$. Let e_o be an index that goes over all the small element edges that make up the big edges O, i.e. $O = \sum_o e_o$. Let nf be an index that goes over all the small element edges that lie on the exterior boundary of the domain and that are not connected to either of the channels. Recall that on such edges, we enforce a no-flow boundary condition, i.e. the normal component of the numerical flux is zero. Recall also that the normal components of the numerical fluxes are continuous on element edges. Thus, summing up (3.27) over all elements in the junction, we obtain:

$$\begin{aligned}\sum_a \int_{\Omega_a} \frac{\partial H^{2D}}{\partial t} \, \mathbf{dx} &= -\sum_E \int_E \hat{\mathbf{Q}}^{2D} \cdot \mathbf{n} \, \mathbf{ds} \\ &= -\sum_i \int_{e_i} \hat{\mathbf{Q}}^{2D} \cdot \mathbf{n} \, \mathbf{ds} - \sum_o \int_{e_o} \hat{\mathbf{Q}}^{2D} \cdot \mathbf{n} \, \mathbf{ds} - \sum_{nf} \int_{nf} \hat{\mathbf{Q}}^{2D} \cdot \mathbf{n} \, \mathbf{ds} \\ &= -\sum_i \int_{e_i} \hat{Q}_n^{2D} \, \mathbf{ds} - \sum_o \int_{e_o} \hat{Q}_n^{2D} \, \mathbf{ds}\end{aligned} \tag{3.28}$$

The local Lax-Friedrichs flux for the 2-D continuity equation is given by the following:

$$\hat{Q}_n = \frac{1}{2}(Q_n^{in} + Q_n^{ex}) + \frac{\lambda}{2}(H^{in} - H^{ex}) \tag{3.29}$$

where, $\lambda = \max\left((|\mathbf{u}^{2D} \cdot \mathbf{n}| + c^{2D})^{in}, (|\mathbf{u}^{2D} \cdot \mathbf{n}| + c^{2D})^{ex}\right)$, $\mathbf{u}^{2D} = \mathbf{Q}^{2D}/H^{2D}$ and $c^{2D} = \sqrt{gH^{2D}}$.

Substituting (3.29) into (3.28), we obtain:

$$\begin{aligned}\sum_a \int_{\Omega_a} \frac{\partial H}{\partial t}^{2D} \mathbf{dx} =& -\frac{1}{2}\sum_i \int_{e_i} \left[Q_{n,i}^{2D,in} + Q_{n,i}^{2D,ex} + \lambda_{e_i}^{2D}(H_i^{2D,in} - H_i^{2D,ex})\right] \mathbf{ds} \\ & -\frac{1}{2}\sum_o \int_{e_o} \left[Q_{n,i}^{2D,in} + Q_{n,i}^{2D,ex} + \lambda_{e_o}^{2D}(H_o^{2D,in} - H_o^{2D,ex})\right] \mathbf{ds} \\ =& -\frac{1}{2}\left[Q_{I,TV}^{2D,in} + \sum_i \lambda_{e_i}^{2D}|e_i|H_{avg,i}^{2D,in} + \sum_i \int_{e_i} (Q_{n,i}^{2D,ex} - \lambda_{e_i}^{2D} H_i^{2D,ex})\ \mathbf{ds}\right] \\ & -\frac{1}{2}\left[Q_{O,TV}^{2D,in} + \sum_o \lambda_{e_o}^{2D}|e_o|H_{avg,o}^{2D,in} + \sum_o \int_{e_o} (Q_{n,o}^{2D,ex} - \lambda_{e_o}^{2D} H_o^{2D,ex})\ \mathbf{ds}\right]\end{aligned} \tag{3.30}$$

Using the interface conditions (3.17a) – (3.17d), we obtain the following expression for the rate of change of water mass in the junction:

$$\begin{aligned}\frac{d}{dt}M_J =& \sum_a \int_{\Omega_a} \frac{\partial H}{\partial t}^{2D} \mathbf{dx} \\ =& -\frac{1}{2}\left[Q_{I,TV}^{2D,in} + \lambda_I \sum_i |e_i| H_{avg,i}^{2D,in} + \sum_i \int_{e_i} \left(-\frac{Q_{\mathbf{s}_{n+\frac{1}{2}}}^{1D,in}}{W_I} - \lambda_I \frac{A_{\mathbf{s}_{n+\frac{1}{2}}}^{1D,in}}{W_I}\right)\ \mathbf{ds}\right] \\ & -\frac{1}{2}\left[Q_{O,TV}^{2D,in} + \lambda_O \sum_o |e_o| H_{avg,o}^{2D,in} + \sum_o \int_{e_o} \left(\frac{Q_{\mathbf{s}_{n+\frac{3}{2}}}^{1D,in}}{W_O} - \lambda_O \frac{A_{\mathbf{s}+\frac{3}{2}}^{1D,in}}{W_O}\right)\ \mathbf{ds}\right] \\ =& -\frac{1}{2}\left[Q_{I,TV}^{2D,in} + \lambda_I \sum_i |e_i| H_{avg,i}^{2D,in} - Q_{\mathbf{s}_{n+\frac{1}{2}}}^{1D,in} - \lambda_I A_{\mathbf{s}_{n+\frac{1}{2}}}^{1D,in}\right] \\ & -\frac{1}{2}\left[Q_{O,TV}^{2D,in} + \lambda_O \sum_o |e_o| H_{avg,o}^{2D,in} + Q_{\mathbf{s}_{n+\frac{3}{2}}}^{1D,in} - \lambda_O A_{\mathbf{s}_{n+\frac{3}{2}}}^{1D,in}\right]\end{aligned} \tag{3.31}$$

Let us now define λ^I and λ^O:

$$\begin{aligned}\lambda^I &= \max(\lambda_{\mathbf{s}_{n+\frac{1}{2}}}^{1D}, \max_i(\lambda_{e_i}^{2D})) \\ \lambda^O &= \max(\lambda_{\mathbf{s}_{n+\frac{3}{2}}}^{1D}, \max_o(\lambda_{e_o}^{2D}))\end{aligned} \tag{3.32}$$

The rate of change of the total water mass in the channel network is given by:

$$\frac{d}{dt}M = \frac{d}{dt}M_c + \frac{d}{dt}M_J \tag{3.33}$$

Substituting expression (3.24) and (3.31) into the above equation, we obtain:

$$\begin{aligned}
\frac{d}{dt}M =& -\frac{1}{2}\left(Q^{1D,in}_{\mathbf{s}_{n+\frac{1}{2}}} - Q^{2D,in}_{I,TV}\right) - \frac{\lambda_I}{2}\left(A^{1D,in}_{\mathbf{s}_{n+\frac{1}{2}}} - \sum_i |e_i| H^{2D,in}_{avg,i}\right) \\
&+ \frac{1}{2}\left(Q^{1D,in}_{\mathbf{s}_{n+\frac{3}{2}}} + Q^{2D,in}_{O,TV}\right) + \frac{\lambda_O}{2}\left(-A^{1D,in}_{\mathbf{s}_{n+\frac{3}{2}}} + \sum_o |e_o| H^{2D,in}_{avg,o}\right) \\
&- \frac{1}{2}\left(Q^{2D,in}_{I,TV} + \lambda_I \sum_i |e_i| H^{2D,in}_{avg,i} - Q^{1D,in}_{\mathbf{s}_{n+\frac{1}{2}}} - \lambda_I A^{1D,in}_{\mathbf{s}_{n+\frac{1}{2}}}\right) \\
&- \frac{1}{2}\left(Q^{2D,in}_{O,TV} + \lambda_O \sum_o |e_o| H^{2D,in}_{avg,o} + Q^{1D,in}_{\mathbf{s}_{n+\frac{3}{2}}} - \lambda_O A^{1D,in}_{\mathbf{s}_{n+\frac{3}{2}}}\right) \\
&= 0
\end{aligned} \tag{3.34}$$

Hence, we obtain that the rate of change of total mass in the channel network with no-flow boundary conditions is zero. This shows that the mass in the entire channel network is conserved.

3.5 Stability of Coupling

In this section, we show that our scheme for coupling channels with junctions results in a stable numerical scheme. We consider a domain consisting of a rectangular channel $C1$ that is connected to a 2-D junction on its outflow end as shown in Figure 3.5. We assume that $C1$ is of unit width, i.e. $W = 1$. For simplicity, we impose a no-flow boundary condition on the inflow

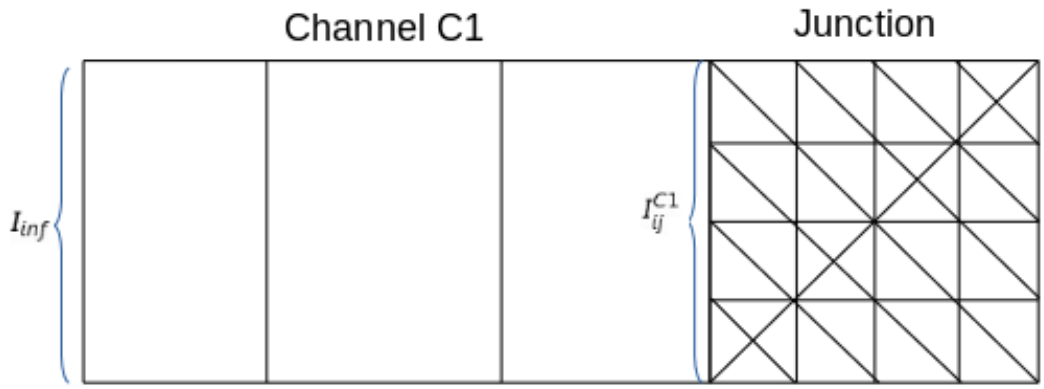


Figure 3.5: Schematic showing a channel connected to a junction on its outflow end

edge of the channel and on all the exterior edges of the junction that are not connected to the channel.

To simplify the stability analysis, we assume that the nonlinear advection terms in the 1-D and the 2-D shallow water equations are negligible and thus we omit them from our equations. We assume that τ_{bf} is constant so that the friction loss is linear with respect to Q. We assume that the topography/bathymetry is constant in the domain and that the channels are prismatic. In order to avoid clutter in notation, we drop the superscripts $1D$ and $2D$ wherever it is obvious with which set of equations we are working. The 1-D shallow water equations that are solved on the channels then become:

$$\begin{aligned} \frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} &= 0 \\ \frac{\partial Q}{\partial t} + \frac{1}{2} g \frac{\partial}{\partial x}\left(A^2\right) &= -g\tau_{bf} Q \end{aligned} \tag{3.35}$$

The 2-D shallow water equations become:

$$\begin{aligned} \frac{\partial H}{\partial t} + \nabla \cdot \mathbf{Q} &= 0 \\ \frac{\partial \mathbf{Q}}{\partial t} + \frac{1}{2} g \nabla H^2 &= -g \tau_{bf} \mathbf{Q} \end{aligned} \tag{3.36}$$

Let Ω_{C1} denote the discretized channel and Ω_J denote the discretized junction. Thus, $\Omega = \Omega_{C1} \cup \Omega_J$ is our simulation domain. Let I_e^{C1} denote the set of indices that go over the elements that discretize the channel. Let I_{int}^{C1} denote the set of indices that go over the interior edges of the channel $C1$, i.e. the nodes at the interface of two elements. Let I_{inf} denote the inflow edge of the channel $C1$. Let I_{ij}^{C1} denote the edge of channel $C1$ connected to the junction. Let $-$ denote the "in" state and $+$ denote the "ex" state. Let $[\![.]\!]$ denote the jump operator defined as: $[\![w]\!] = w^- - w^+$. Now, let $A(x,t),\ Q(x,t),\ \psi(x),\ \phi(x) \in \mathcal{P}^p(\Omega_e)$ for all Ω_e and $t \in [0,T]$. Let Π denote the $L2-$projection operator onto $\bigoplus_e \mathcal{P}^p(\Omega_e)$. Let the initial conditions $A(x,0)$ and $Q(x,0)$ be given by the $L2-$projections of the initial conditions prescribed for the equations given in (3.35). Let Q_B and A_B denote the prescribed values for Q and A respectively at the boundary. The solution $(A(x,t), Q(x,t))$ satisfies the following global semi-

discrete weak formulation of the 1-D shallow water equations for channel $C1$:

$$\sum_{e\in I_e^{C1}} \left\langle \frac{\partial A}{\partial t}, \psi \right\rangle_{\Omega_e} + \sum_{i\in I_{int}^{C1}} \hat{Q}[\![\psi]\!] - Q_B\psi^+|_{I_{inf}} + \hat{Q}\psi^-|_{I_{ij}^{C1}} - \sum_{e\in I_e^{C1}} \left\langle Q, \frac{\partial \phi}{\partial x} \right\rangle_{\Omega_e} = 0 \tag{3.37}$$

$$\sum_{e\in I_e^{C1}} \left\langle \frac{\partial Q}{\partial t}, \psi \right\rangle_{\Omega_e} + \sum_{i\in I_{int}^{C1}} \left(\widehat{\frac{g\Pi A^2}{2}} \right) [\![\phi]\!] - \left. \frac{g\Pi A_B^2}{2}\phi^+ \right|_{I_{inf}} + \left. \widehat{\frac{g\Pi A^2}{2}}\phi^- \right|_{I_{ij}^{C1}}$$
$$- \sum_{e\in I_e^{C1}} \left\langle \left(\frac{g\Pi A^2}{2} \right), \frac{\partial \phi}{\partial x} \right\rangle_{\Omega_e} + \sum_{e\in I_e^{C1}} \langle g\tau_{bf}Q, \phi \rangle_{\Omega_e} = 0 \tag{3.38}$$

Let I_e^J denote the set of indices that go over all the triangular elements that discretize the junction. Let I_{int}^J denote the set of indices that go over all the interior edges of the junction. Let I_{ext}^{ch} denote the set of indices that go over all the exterior edges of the junction that are connected to the channel, so that $\sum_{i\in I_{ext}^{ch}} e_i = I_{ij}^{C1}$. Let I_{nf} denote the set of indices that go over all the exterior edges of the junction that are not connected to the channel. Let the initial conditions $H(\mathbf{x}, 0)$ and $\mathbf{Q}(\mathbf{x}, 0)$ be given by the $L2-$ projections of the initial conditions prescribed for the equations given in (3.36). Let $\mathbf{Q}_B$ and H_B denote the prescribed values for $\mathbf{Q}\cdot\mathbf{n}$ and H respectively at the boundary. The solution $(H(\mathbf{x}, t), \mathbf{Q}(\mathbf{x}, t))$ satisfies the following global semi-discrete weak

formulation of the 2-D shallow water equations:

$$\sum_{e\in I_e^J}\left(\frac{\partial H}{\partial t},\psi\right)_{\Omega_e}+\sum_{i\in I_{int}^J}\left\langle\widehat{\mathbf{Q}\cdot\mathbf{n}},[\![\psi]\!]\right\rangle_{e_i}+\sum_{i\in I_{nf}}\left\langle\mathbf{Q}_B\cdot\mathbf{n},\psi^-\right\rangle_{e_i}+\sum_{i\in I_{ext}^{ch}}\left\langle\widehat{\mathbf{Q}\cdot\mathbf{n}},\psi^-\right\rangle_{e_i}$$
$$-\sum_{e\in I_e^J}(\mathbf{Q},\nabla\psi)_{\Omega_e}=0 \tag{3.39}$$

$$\sum_{e\in I_e^J}\left(\frac{\partial\mathbf{Q}}{\partial t},\boldsymbol{\phi}\right)_{\Omega_e}+\sum_{i\in I_{int}^J}\left\langle\widehat{\frac{g\Pi H^2}{2}}\mathbf{n},[\![\boldsymbol{\phi}]\!]\right\rangle_{e_i}+\sum_{i\in I_{nf}}\left\langle\frac{g\Pi H_B^2}{2}\mathbf{n},[\![\boldsymbol{\phi}]\!]\right\rangle_{e_i}$$
$$\sum_{i\in I_{ext}^{ch}}\left\langle\widehat{\frac{g\Pi H^2}{2}}\mathbf{n},\boldsymbol{\phi}\right\rangle_{e_i}-\sum_{e\in I_e^J}\left(\frac{g\Pi H^2}{2},\nabla\cdot\boldsymbol{\phi}\right)_{\Omega_e}+\sum_{e\in I_e^J}(g\tau_{bf}\mathbf{Q},\boldsymbol{\phi})_{\Omega_e}=0 \tag{3.40}$$

We use the local Lax–Friedrichs flux as the numerical flux on the interior edges. However, we make a small modification in the stabilization term in the continuity equation. The numerical fluxes are given below:

$$\forall i\in I_{int}^{C1},\qquad \hat{Q}_i=\{\!\{Q\}\!\}+\frac{\lambda_i^{1D}}{2}\left[\!\!\left[\frac{g\Pi A^2}{2}\right]\!\!\right] \tag{3.41}$$

$$\forall i\in I_{int}^{C1},\qquad \left(\widehat{\frac{g\Pi A^2}{2}}\right)_i=\left\{\!\!\left\{\frac{g\Pi A^2}{2}\right\}\!\!\right\}+\frac{\lambda_i^{1D}}{2}[\![Q]\!] \tag{3.42}$$

$$\forall i\in I_{int}^{J},\qquad \widehat{\mathbf{Q}\cdot\mathbf{n}}_i=\{\!\{\mathbf{Q}\cdot\mathbf{n}\}\!\}+\frac{\lambda_i^{2D}}{2}\left[\!\!\left[\frac{g\Pi H^2}{2}\right]\!\!\right] \tag{3.43}$$

$$\forall i\in I_{int}^{J},\qquad \left(\widehat{\frac{g\Pi H^2}{2}}\right)_i=\left\{\!\!\left\{\frac{g\Pi H^2}{2}\right\}\!\!\right\}+\frac{\lambda_i^{2D}}{2}[\![\mathbf{Q}\cdot\mathbf{n}]\!] \tag{3.44}$$

where

$$\lambda^{1D}=\max(|u^{1D}|^+ + c^{1D},|u^{1D}|^- + c^{1D})$$

and

$$\lambda^{2D}=\max(|\mathbf{u}\cdot\mathbf{n}|^+ + c^{2D},|\mathbf{u}\cdot\mathbf{n}|^- + c^{2D}).$$

We use the average flux as the numerical flux on the interface edges of the channel and junction as follows:

$$\forall i \in I_{ij}^{C1}, \qquad \hat{Q}_i = \{\!\{Q\}\!\} \tag{3.45}$$

$$\forall i \in I_{ij}^{C1}, \qquad \left(\widehat{\frac{g\Pi A^2}{2}}\right)_i = \left\{\!\!\left\{\frac{g\Pi A^2}{2}\right\}\!\!\right\} \tag{3.46}$$

$$\forall i \in I_{ext}^{ch}, \qquad \widehat{\mathbf{Q}\cdot\mathbf{n}}_i = \{\!\{\mathbf{Q}\cdot\mathbf{n}\}\!\} \tag{3.47}$$

$$\forall i \in I_{ext}^{ch}, \qquad \left(\widehat{\frac{g\Pi H^2}{2}}\right)_i = \left\{\!\!\left\{\frac{g\Pi H^2}{2}\right\}\!\!\right\} \tag{3.48}$$

Lemma 3.5.1 (Discrete stability of coupling)**.** *Let $H^{1D}, H^{2D} > 0$ in Ω. Then, assuming a no-flow boundary condition on $\partial\Omega$, the following stability result holds for the numerical scheme* (3.37) − (3.48)*:*

$$\begin{aligned}&\frac{g}{3}\|A(T)\|^3_{L^3(\Omega_{C1})} + \|Q(T)\|^2_{L^2(\Omega_{C1})} + \frac{g}{3}\|H(T)\|^3_{L^3(\Omega_J)} + \|\mathbf{Q}(T)\|^2_{L^2(\Omega_J)} \\ &\leq \frac{g}{3}\|A(0)\|^3_{L^3(\Omega_{C1})} + \|Q(0)\|^2_{L^2(\Omega_{C1})} + \frac{g}{3}\|H(0)\|^3_{L^3(\Omega_J)} + \|\mathbf{Q}(0)\|^2_{L^2(\Omega_J)}\end{aligned} \tag{3.49}$$

Proof. Substituting $\psi = \frac{g\Pi A^2}{2}$ and $\phi = Q$ in (3.37) and (3.38) and adding them together, we obtain:

$$\underbrace{\sum_{e\in I_e^{C1}}\left\langle\frac{\partial A}{\partial t},\frac{g\Pi A^2}{2}\right\rangle_{\Omega_e}}_{T_1}+\underbrace{\sum_{i\in I_{int}^{C1}}\hat{Q}\left[\!\left[\frac{g\Pi A^2}{2}\right]\!\right]}_{T_2}-\underbrace{Q_B\left(\frac{g\Pi A^2}{2}\right)^+\Big|_{I_{inf}}}_{T_3}$$

$$+\underbrace{\hat{Q}\left(\frac{g\Pi A^2}{2}\right)^-\Big|_{I_{ij}^{C1}}}_{T_4}-\underbrace{\sum_{e\in I_e^{C1}}\left\langle Q,\frac{\partial}{\partial x}\left(\frac{g\Pi A^2}{2}\right)\right\rangle_{\Omega_e}}_{T_5}$$

$$+\underbrace{\sum_{e\in I_e^{C1}}\left\langle\frac{\partial Q}{\partial t},Q\right\rangle_{\Omega_e}}_{T_6}+\underbrace{\sum_{i\in I_{int}^{C1}}\left(\widehat{\frac{g\Pi A^2}{2}}\right)[\![Q]\!]}_{T_7}-\underbrace{\left(\frac{g\Pi A_B^2}{2}\right)Q^+\Big|_{I_{inf}}}_{T_8}+\underbrace{\left(\widehat{\frac{g\Pi A^2}{2}}\right)Q^-\Big|_{I_{ij}^{C1}}}_{T_9}$$

$$-\underbrace{\sum_{e\in I_e^{C1}}\left\langle\frac{g\Pi A^2}{2},\frac{\partial Q}{\partial x}\right\rangle_{\Omega_e}}_{T_{10}}+\underbrace{\sum_{e\in I_e^{C1}}\langle g\tau_{bf}Q,Q\rangle_{\Omega_e}}_{T_{11}}=0 \tag{3.50}$$

Now, we will expand the numerical flux terms. Using the numerical flux described in (3.41) and (3.42), we obtain:

$$T_2=\sum_{i\in I_{int}^{C1}}\hat{Q}\left[\!\left[\frac{g\Pi A^2}{2B}\right]\!\right]=\underbrace{\sum_{i\in I_{int}^{C1}}\{\!\{Q\}\!\}\left[\!\left[\frac{g\Pi A^2}{2}\right]\!\right]}_{T_{21}}+\underbrace{\sum_{i\in I_{int}^{C1}}\frac{g\lambda_i}{4}[\![\Pi A^2]\!][\![\Pi A^2]\!]}_{T_{22}}$$

$$T_7=\sum_{i\in I_{int}^{C1}}\left(\widehat{\frac{g\Pi A^2}{2}}\right)[\![Q]\!]=\underbrace{\sum_{i\in I_{int}^{C1}}\left\{\!\!\left\{\frac{g\Pi A^2}{2}\right\}\!\!\right\}[\![Q]\!]}_{T_{71}}+\underbrace{\sum_{i\in I_{int}^{C1}}\frac{\lambda_i}{2}[\![Q]\!][\![Q]\!]}_{T_{72}} \tag{3.51a}$$

Now take a look at the following term:

$$-T_5 \equiv -\sum_{e\in I_e^{C1}} \left\langle Q, \frac{\partial}{\partial x}\left(\frac{g\Pi A^2}{2}\right)\right\rangle_{\Omega_e} = \underbrace{\sum_{e\in I_e^{C1}} \left\langle \frac{\partial Q}{\partial x}, \frac{g\Pi A^2}{2}\right\rangle_{\Omega_e}}_{T_{51}} - \underbrace{\sum_{i\in I_{int}^{C1}} \left[\!\left[Q\left(\frac{g\Pi A^2}{2}\right)\right]\!\right]}_{T_{52}}$$
$$\underbrace{- Q^-\left(\frac{g\Pi A^2}{2}\right)^-\Big|_{I_{ij}}}_{T_{53}} + \underbrace{Q^+\left(\frac{g\Pi A^2}{2}\right)^+\Big|_{I_{inf}}}_{T_{54}} \tag{3.51b}$$

Using the identity $[\![uv]\!] = \{\!\{u\}\!\}[\![v]\!] + [\![u]\!]\{\!\{v\}\!\}$, we can decompose T_{52} further:

$$-T_{52} = -\sum_{i\in I_{int}^{C1}} \left[\!\left[Q\left(\frac{g\Pi A^2}{2}\right)\right]\!\right] = \underbrace{-\sum_{i\in I_{int}^{C1}} \{\!\{Q\}\!\} \left[\!\left[\frac{g\Pi A^2}{2}\right]\!\right]}_{T_{521}} \underbrace{-\sum_{i\in I_{int}^{C1}} [\![Q]\!] \left\{\!\!\left\{\frac{g\Pi A^2}{2}\right\}\!\!\right\}}_{T_{522}} \tag{3.51c}$$

Let us now look at the inflow boundary terms. Our no-flow boundary condition is imposed in the following manner:

$$Q_B|_{I_{inf}} = 0$$
$$g\Pi A_B^2\big|_{I_{inf}} = g(\Pi A^2)^+\big|_{I_{inf}}$$

Using these boundary conditions, we get:

$$T_3 = 0$$
$$T_8 = \left(\frac{g\Pi A^2}{2}\right)^+ Q^+\Big|_{I_{inf}} = T_{54} \tag{3.51d}$$

Let us consider the terms arising on the channel edge connected to the junction.

Using the interface flux defined in (3.45) and (3.46), we get:

$$
\begin{aligned}
T_4 &= \hat{Q}\left(\frac{g\Pi A^2}{2}\right)^-\Bigg|_{I_{ij}^{C1}} \\
&= \underbrace{\frac{Q^-}{2}\left(\frac{g\Pi A^2}{2}\right)^-\Bigg|_{I_{ij}^{C1}}}_{T_{41}} + \underbrace{\frac{Q^+}{2}\left(\frac{g\Pi A^2}{2}\right)^-\Bigg|_{I_{ij}^{C1}}}_{T_{42}} \\
T_9 &= \widehat{\left(\frac{g\Pi A^2}{2}\right)}Q^-\Bigg|_{I_{ij}^{C1}} \\
&= \underbrace{\left(\frac{g\Pi A^2}{4}\right)^- Q^-\Bigg|_{I_{ij}^{C1}}}_{T_{91}} + \underbrace{\left(\frac{g\Pi A^2}{4}\right)^+ Q^-\Bigg|_{I_{ij}^{C1}}}_{T_{92}}
\end{aligned}
\tag{3.51e}
$$

Now, let us look at the terms with the time derivative.

$$
\begin{aligned}
T_1 = \sum_{e\in I_e^{C1}}\left\langle\frac{\partial A}{\partial t},\frac{g\Pi A^2}{2}\right\rangle_{\Omega_e} &= \sum_{e\in I_e^{C1}}\int_{\Omega_e}\frac{g}{2}\frac{\partial A}{\partial t}A^2\ \mathbf{ds} \\
&= \frac{g}{2}\sum_{e\in I_e^{C1}}\int_{\Omega_e}\frac{1}{3}\frac{\partial}{\partial t}\left(A\right)^3\ \mathbf{ds} \\
&= \frac{g}{6}\frac{d}{dt}\sum_{e\in I_e^{C1}}\int_{\Omega_e}|A|^3\ \mathbf{ds} \quad [A>0] \\
&= \frac{g}{6}\frac{d}{dt}\sum_{e\in I_e^{C1}}\|A\|^3_{L^3(\Omega_e)} \\
&= \frac{g}{6}\frac{d}{dt}\|A\|^3_{L^3(\Omega_{C1})}
\end{aligned}
\tag{3.51f}
$$

Similarly,

$$
T_6 = \sum_{e\in I_e^{C1}}\left\langle\frac{\partial Q}{\partial t},Q\right\rangle_{\Omega_e} = \frac{1}{2}\frac{d}{dt}\|Q\|^2_{L^2(\Omega_{C1})} \tag{3.51g}
$$

Finally, we look at the bottom friction term:

$$
T_{11} = \sum_{e\in I_e^{C1}} < g\tau_{bf}Q,Q >_{\Omega_e} = g\tau_{bf}\|Q\|^2_{L^2(\Omega_{C1})} \tag{3.51h}
$$

Notice that we now have the following relationships:

$$\begin{aligned} T_{21} - T_{521} &= 0 \\ T_{71} - T_{522} &= 0 \\ T_{51} - T_{10} &= 0 \\ T_{54} - T_{8} &= 0 \\ T_{41} + T_{91} - T_{53} &= 0 \end{aligned} \tag{3.51i}$$

Using (3.51a)–(3.51i), we can reduce (3.50) to the following form:

$$\begin{aligned} &\frac{g}{6}\frac{d}{dt}\left\|A\right\|^3_{L^3(\Omega_{C1})} + \frac{1}{2}\frac{d}{dt}\left\|Q\right\|^2_{L^2(\Omega_{C1})} + \sum_{i\in I^{C1}_{int}} \frac{g\lambda_i}{4}\left[\!\left[\Pi A^2\right]\!\right]^2 + \sum_{i\in I^{C1}_{int}} \frac{\lambda_i}{2}[\![Q]\!]^2 \\ &+ g\tau_{bf}\left\|Q\right\|^2_{L^2(\Omega_{C1})} + \frac{Q^+}{2}\left(\frac{g\Pi A^2}{2}\right)^-\Bigg|_{I^{C1}_{ij}} + \left(\frac{g\Pi A^2}{4}\right)^+ Q^-\Bigg|_{I^{C1}_{ij}} = 0 \end{aligned} \tag{3.52}$$

Now, let us turn our attention to the 2-D shallow water equations that are solved in the junction. Substituting $\psi = \frac{g\Pi H^2}{2}$ and $\boldsymbol{\phi} = \mathbf{Q}$ into (3.39) and (3.40) and adding them together, we obtain:

$$\begin{aligned} &\sum_{e\in I^J_e}\left(\frac{\partial H}{\partial t}, \frac{g\Pi H^2}{2}\right)_{\Omega_e} + \sum_{i\in I^J_{int}}\left\langle \widehat{\mathbf{Q}\cdot\mathbf{n}}, \left[\!\left[\frac{g\Pi H^2}{2}\right]\!\right]\right\rangle_{e_i} + \sum_{i\in I_{nf}}\left\langle \mathbf{Q}_B\cdot\mathbf{n}, \frac{g\Pi H^2}{2}^-\right\rangle_{e_i} \\ &+ \sum_{i\in I^{ch}_{ext}}\left\langle \widehat{\mathbf{Q}\cdot\mathbf{n}}, \frac{g\Pi H^2}{2}^-\right\rangle_{e_i} - \sum_{e\in I^J_e}\left(\mathbf{Q}, \nabla\frac{g\Pi H^2}{2}\right)_{\Omega_e} \\ &\sum_{e\in I^J_e}\left(\frac{\partial \mathbf{Q}}{\partial t}, \mathbf{Q}\right)_{\Omega_e} + \sum_{i\in I^J_{int}}\left\langle \widehat{\frac{g\Pi H^2}{2}}\mathbf{n}, [\![\mathbf{Q}]\!]\right\rangle_{e_i} + \sum_{i\in I_{nf}}\left\langle \frac{g\Pi H_B^2}{2}\mathbf{n}, [\![\mathbf{Q}]\!]\right\rangle_{e_i} \\ &\sum_{i\in I^{ch}_{ext}}\left\langle \widehat{\frac{g\Pi H^2}{2}}\mathbf{n}, \mathbf{Q}\right\rangle_{e_i} - \sum_{e\in I^J_e}\left(\frac{g\Pi H^2}{2}, \nabla\cdot\mathbf{Q}\right)_{\Omega_e} + \sum_{e\in I^J_e}(g\tau_{bf}\mathbf{Q}, \mathbf{Q})_{\Omega_e} = 0 \end{aligned} \tag{3.53}$$

Using the no-flow boundary conditions and performing the same manipulations as in the case of the 1-D shallow water equations, we obtain the following simplified form of the equation above:

$$\frac{g}{6}\frac{d}{dt}\left\|H\right\|^3_{L^3(\Omega_J)} + \frac{1}{2}\left\|\mathbf{Q}\right\|^2_{L^2(\Omega_J)} + \sum_{i\in I^J_{int}} \frac{g\lambda_i}{4}\left[\!\left[\Pi H^2\right]\!\right]^2 + \sum_{i\in I^J_{int}} \frac{\lambda i}{2}[\![\mathbf{Q}]\!]\cdot[\![\mathbf{Q}]\!]$$
$$+ g\tau_{bf}\left\|\mathbf{Q}\right\|^2_{L^2(\Omega_J)} + \sum_{i\in I^{ch}_{ext}}\int_{e_i}\frac{g}{4}Q_n^+(\Pi H^2)^-\ \mathbf{ds} + \sum_{i\in I^{ch}_{ext}}\int_{e_i}\left(\frac{g\Pi H^2}{4}\right)^+ Q_n^-\ \mathbf{ds} = 0 \tag{3.54}$$

Now, we add (3.52) and (3.54) to obtain the following equation for the entire domain:

$$\frac{g}{6}\frac{d}{dt}\left\|A\right\|^3_{L^3(\Omega_{C1})} + \frac{1}{2}\frac{d}{dt}\left\|Q\right\|^2_{L^2(\Omega_{C1})} + \frac{g}{6}\frac{d}{dt}\left\|H\right\|^3_{L^3(\Omega_J)} + \frac{1}{2}\left\|\mathbf{Q}\right\|^2_{L^2(\Omega_J)}$$
$$+ \underbrace{\sum_{i\in I^{C1}_{int}} \frac{g\lambda_i}{4}\left[\!\left[\Pi A^2\right]\!\right]^2}_{R_1} + \underbrace{\sum_{i\in I^{C1}_{int}} \frac{\lambda_i}{2}[\![Q]\!]^2}_{R_2} + \underbrace{\sum_{i\in I^J_{int}} \frac{g\lambda_i}{4}\left[\!\left[\Pi H^2\right]\!\right]^2}_{R_3} + \underbrace{\sum_{i\in I^J_{int}} \frac{\lambda i}{2}[\![\mathbf{Q}]\!]\cdot[\![\mathbf{Q}]\!]}_{R_4}$$
$$+ \underbrace{g\tau_{bf}\left\|Q\right\|^2_{L^2(\Omega_{C1})}}_{R_5} + \underbrace{g\tau_{bf}\left\|\mathbf{Q}\right\|^2_{L^2(\Omega_J)}}_{R_6} + \underbrace{\frac{Q^+}{2}\left(\frac{g\Pi A^2}{2}\right)^-\Bigg|_{I^{C1}_{ij}}}_{R_7}$$
$$+ \underbrace{\left(\frac{g\Pi A^2}{4}\right)^+ Q^-\Bigg|_{I^{C1}_{ij}}}_{R_8} + \underbrace{\sum_{i\in I^{ch}_{ext}}\int_{e_i}\frac{g}{4}Q_n^+(\Pi H^2)^-\ \mathbf{ds}}_{R_9} + \underbrace{\sum_{i\in I^{ch}_{ext}}\int_{e_i}\left(\frac{g\Pi H^2}{4}\right)^+ Q_n^-\ \mathbf{ds}}_{R_{10}} = 0 \tag{3.55}$$

Because the interface conditions do not define $(\Pi H^2)^+$ and $(\Pi A^2)^+$ at

the interface, let us do that below:

$$\begin{aligned}(\Pi H^2)^+_{2D} &= \Pi \left(\frac{A^2}{B^2}\right)^-_{1D} \\ (\Pi A^2)^+_{1D} &= \sum_{i\in I^{ch}_{ext}} \int_{e_i} (\Pi H^2)^-_{2D}\ \mathbf{ds}\end{aligned} \tag{3.56}$$

Now, using the definitions above and the interface conditions (3.17a) – (3.17d), we can manipulate some terms further as shown below:

$$\begin{aligned}R_7 &= \frac{Q^+}{2}\left(\frac{g\Pi A^2}{2}\right)^-\Big|_{I^{c1}_{ij}} \\ &= -\frac{g}{4}\left(\sum_{i\in I^{ch}_{ext}} \int_{e_i} Q^-_n\ \mathbf{ds}\right)\left(\Pi A^2\right)^-\Big|_{I^{C1}_{ij}} \qquad (3.57)\\ R_{10} &= \sum_{i\in I^{ch}_{ext}} \int_{e_i} \left(\frac{g\Pi H^2}{4}\right)^+ Q^-_n\ \mathbf{ds} \\ &= \frac{g}{4}\sum_{i\in I^{ch}_{ext}} \int_{e_i} Q^-_n\ \Pi\left[\left(\frac{A}{B}\right)^2\right]^-\Big|_{I^{C1}_{ij}}\ \mathbf{ds} \\ &= \frac{g}{4}\left(\sum_{i\in I^{ch}_{ext}} \int_{e_i} Q^-_n\ \mathbf{ds}\right)(\Pi A^2)^-\big|_{I^{C1}_{ij}}\,\mathbf{ds} \qquad (3.58)\\ R_8 &= \left(\frac{g\Pi A^2}{4}\right)^+ Q^-\Big|_{I^{C1}_{ij}} \\ &= \frac{g}{4}\left(\sum_{i\in I^{ch}_{ext}} \int_{e_i} (\Pi H^2)^-\mathbf{ds}\right) Q^-\Big|_{I^{C1}_{ij}} \qquad (3.59)\end{aligned}$$

$$
\begin{aligned}
R_9 &= \sum_{i \in I_{ext}^{ch}} \int_{e_i} \frac{g}{4} Q_n^+ (\Pi H^2)^- \mathbf{ds} \\
&= -\frac{g}{4} \sum_{i \in I_{ext}^{ch}} \int_{e_i} \frac{Q^-}{B} \Big|_{I_{ij}^{C1}} (\Pi H^2)^- \mathbf{ds} \\
&= -\frac{g}{4} \left(\sum_{i \in I_{ext}^{ch}} \int_{e_i} (\Pi H^2)^- \mathbf{ds} \right) Q^- \Big|_{I_{ij}^{C1}}
\end{aligned} \tag{3.60}
$$

Thus we get the following relationships:

$$
\begin{aligned}
R_7 + R_{10} &= 0 \\
R_8 + R_9 &= 0
\end{aligned} \tag{3.61}
$$

Substituting (3.61) into (3.55) and noticing that terms $R_1 - R_4$ are positive, we obtain:

$$
\frac{g}{6} \frac{d}{dt} \|A\|^3_{L^3(\Omega_{C1})} + \frac{1}{2} \frac{d}{dt} \|Q\|^2_{L^2(\Omega_{C1})} + \frac{g}{6} \frac{d}{dt} \|H\|^3_{L^3(\Omega_J)} + \frac{1}{2} \frac{d}{dt} \|\mathbf{Q}\|^2_{L^2(\Omega_J)} \leq 0 \tag{3.62}
$$

As a final step, we integrate (3.62) from time 0 to T:

$$
\begin{aligned}
&\frac{g}{6} \int_0^T \frac{d}{dt} \|A\|^3_{L^3(\Omega_{C1})} \, \mathrm{dt} + \frac{1}{2} \int_0^T \frac{d}{dt} \|Q\|^2_{L^2(\Omega_{C1})} \, \mathrm{dt} + \frac{g}{6} \int_0^T \frac{d}{dt} \|H\|^3_{L^3(\Omega_J)} \, \mathrm{dt} \\
&+ \frac{1}{2} \int_0^T \frac{d}{dt} \|\mathbf{Q}\|^2_{L^2(\Omega_J)} \, \mathrm{dt} \leq 0
\end{aligned} \tag{3.63}
$$

This gives us the following stability result for the entire channel network:

$$
\begin{aligned}
&\frac{g}{3} \|A(T)\|^3_{L^3(\Omega_{C1})} + \|Q(T)\|^2_{L^2(\Omega_{C1})} + \frac{g}{3} \|H(T)\|^3_{L^3(\Omega_J)} + \|\mathbf{Q}(T)\|^2_{L^2(\Omega_J)} \\
&\leq \frac{g}{3} \|A(0)\|^3_{L^3(\Omega_{C1})} + \|Q(0)\|^2_{L^2(\Omega_{C1})} + \frac{g}{3} \|H(0)\|^3_{L^3(\Omega_J)} + \|\mathbf{Q}(0)\|^2_{L^2(\Omega_J)}
\end{aligned} \tag{3.64}
$$

□

3.6 Numerical Tests for Channel Network

In this section, we test our numerical scheme for simulating flow in a network of channels on a number of problems.

3.6.1 Shock propagation

Here, we show that shocks are properly transmitted through the interfaces of channel segments and junctions via a numerical simulation. This test shows that the coupling of the channels with junctions does not violate the Rankine–Hugoniot condition. We use a test case that was developed in [35] to test the effects of boundary conditions on the shock waves. The test problem consists of a flat, frictionless rectangular channel 1000 m long and 1 m wide. Water in the channel is initially at rest at 2 m high. We supply a constant discharge of 140 m^3/s at the upstream end of the channel. After the introduction of the discharge at the upstream end, the water height at the upstream end is kept constant at 12 m. These conditions introduce a shock wave in the channel.

We run two different simulations for this test problem. In the first simulation, we discretize the entire rectangular channel as a 1-D channel with $h = 2$ m. In the second simulation, we decompose the rectangular channel into two channels that are connected to each other via a small junction that extends from $x = 400$ to $x = 410$. This decomposition is shown in Figure 3.6, where two interfaces exist between the channels and the junction at $x = 400$ and at $x = 410$. The junction is discretized with an unstructured mesh with

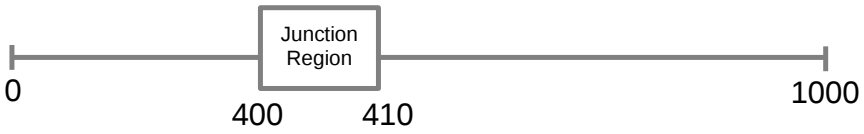


Figure 3.6: Schematic of the decomposition of the rectangular channel for the shock propagation test

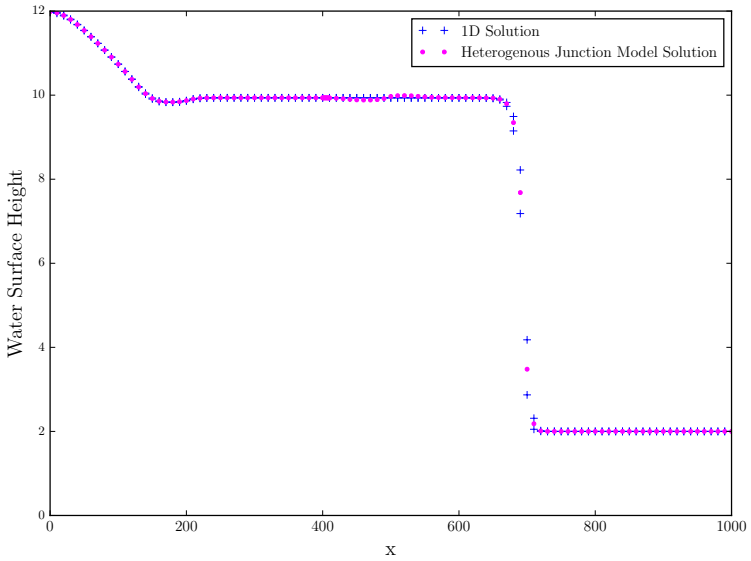


Figure 3.7: Water surface height obtained from the 1-D model and the channel network model

60 elements with $h_{max} = .5$ m. The channels are discretized with $h = 10$ m.

In Figure 3.7, we have plotted the water surface height obtained from the channel network model simulation against the water surface height obtained from the 1-D model simulation at time $t = 40.5$ s. The figure shows a clear agreement between the results of these two simulations. Observe that at time $t = 40.5$ s, the channel network model simulation and the 1-D simulation both have a shock at the same location. In the channel network model simulation, this shock has passed through the two channel-junction interfaces and is moving at exactly the same speed as in the 1-D simulation. These observations verify that the interface conditions (3.17a) – (3.17d) allow for the

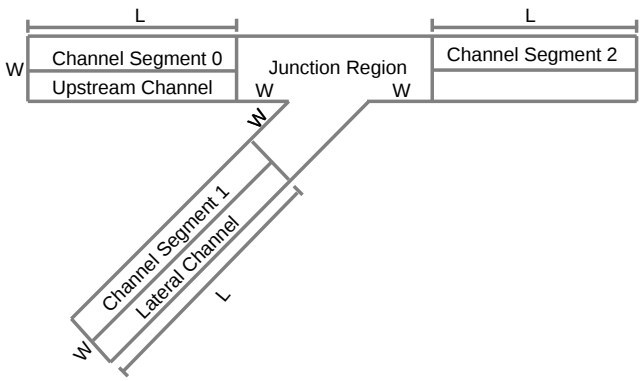


Figure 3.8: A schematic of the domain decomposition for the 45° junction

proper transmission of shocks through the interfaces of channel segments and junctions.

3.6.2 Comparison with other models

In this section, we compare our channel network model with other junction models. The test case involves two channels that combine at a 45° angle as shown in Figure 3.8. The channels all have a uniform width of $W = 10$ m and the junction extends W m into each channel. Each channel has a length of $L = 600$ m and a bottom slope equal to 0.001. The channels all slope in the direction of the flow. The domain with the bathymetry is shown in Figure 3.9. The channels also have friction. The Manning's n values for channels 0, 1 and 2 are 0.0141, 0.0138 and 0.0125 respectively. Water enters the system through channels 0 and 1 or the upstream and the lateral channels respectively. We supply a constant discharge of 30 m^3/s to channel 0 and a constant discharge of 20 m^3/s to channel 1. We keep the water height constant at 1.69 m at the end of channel 2 or the downstream channel. For the channel network

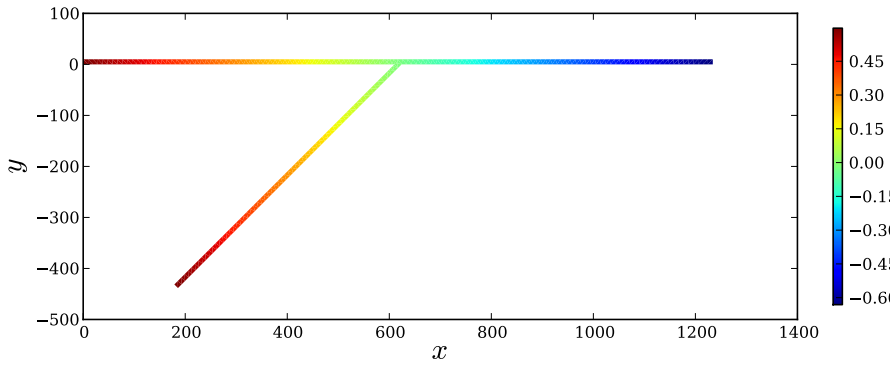
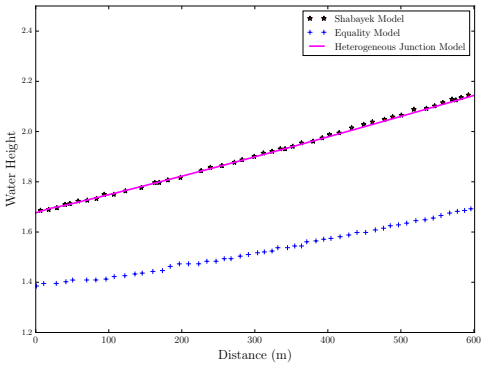


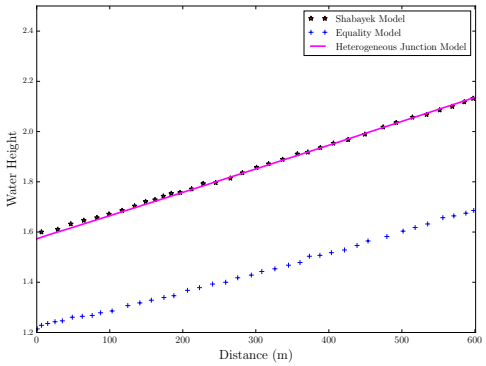
Figure 3.9: Domain for simulating flow through a 45° junction with bathymetry information

model, the channels are each 600 m long and the junction extends W m in the direction of each channel. We discretize the channels with a uniform $h = 20$ m and the junction with 24 elements with $h_{max} = 20$ m.

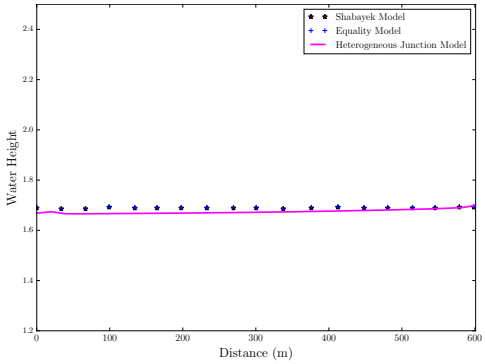
This case was studied in [51] for comparing different 1-D junction models found in the literature. The two junction models to which we compare our channel network model are the "Equality Model" and the "Shabayek Model" described in [51]. The results from our heterogeneous junction model along with other junction models are presented in Figure 3.10. Instead of implementing these models ourselves, we extract the "Equality Model" and the "Shabayek Model" data presented in [51] using WebPlotDigitizer [78] and plot our results against the data thus extracted. Note that the h value we use for our simulations are the same as the h value used in [51].



(a) Water height in Channel 0



(b) Water height in Channel 1



(c) Water height in Channel 2

Figure 3.10: Comparison of water height obtained from different junction models with our channel network model

As we can see in Figure 3.10, the results produced by our channel network model agree closely with the results produced by the "Shabayek Model." By comparing the results of different junction models with experimental data and calculating the relative error, the "Shabayek Model" was determined to be the most accurate model in [51]. This establishes the accuracy of our channel network model. As mentioned previously, our results were produced using a

very small 2-D junction discretized with a small number of elements. Thus, the computational cost of the model is not too high. Comparison of the cost of implementing our model vs non-linear models like the "Shabayek Model" remains a topic of future study.

3.6.3 Flow through a 90° junction

In this section, we validate our channel network model using experimental data reported in [93]. The experiment studied flow through a sharp-edged 90° junction. Our goal is to verify the capability of our channel network model to reproduce some prominent characteristics of such flows which include a zone of separation immediately downstream of the junction lateral channel, a contracted flow region in the main channel due to the separation zone and a possible stagnation point immediately upstream of the junction [93].

In the experimental design, the main channel and the lateral channels are 0.914 m wide. The main channel is 21.95 m long and the lateral channel is 3.66 m long. The lateral channel meets with the main channel 5.49 m downstream from the main channel's boundary (the left end). A constant fully developed flow is supplied at the left (upstream) end of the main channel and at the opening of the lateral channel. A constant water height is maintained at the downstream end. The coordinates are normalized by the channel width, W, resulting in the non-dimensional coordinates, x^* and y^*. The water surface height is also normalized by W, and is reported as the non-dimensional variable ζ^*. The velocity measurement is non-dimensionalized by the downstream av-

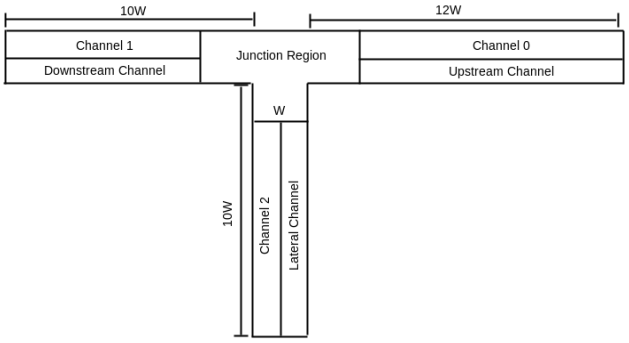


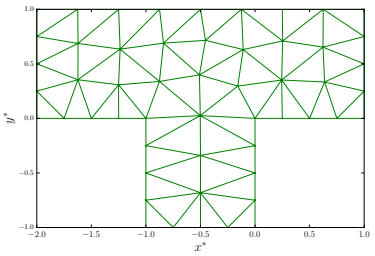
Figure 3.11: A schematic of the domain decomposition for the 90° junction

erage velocity, $V = 0.628$ m/s, and is reported as the non-dimensional velocity u^*.

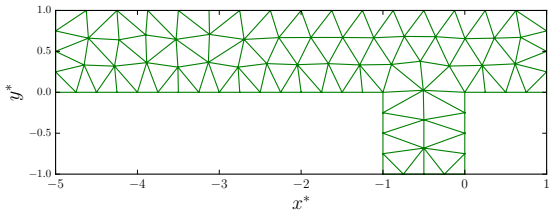
In [93], different experiments with varying flow ratio q^* are performed. q^* is defined as the ratio of the upstream main channel flow to the total flow. The total flow for all of the experiments is 0.170 m^3/s. We choose the case with $q^* = 0.250$, which means that the discharge coming in through the lateral channel is higher than the discharge coming in through the main channel. The momentum of the flow coming in from the lateral channel causes the development of a larger zone of separation and a stagnation point. A constant discharge of 0.042 m^3/s is supplied at the upstream end of the main channel, and a discharge of 0.127 m^3/s is supplied at the upstream end of the lateral channel. A constant water height of 0.296 m is enforced at the downstream end of the main channel.

For the simulation, the domain has been chosen as suggested in [64] in order to ensure that fully developed flows are obtained at the upstream ends

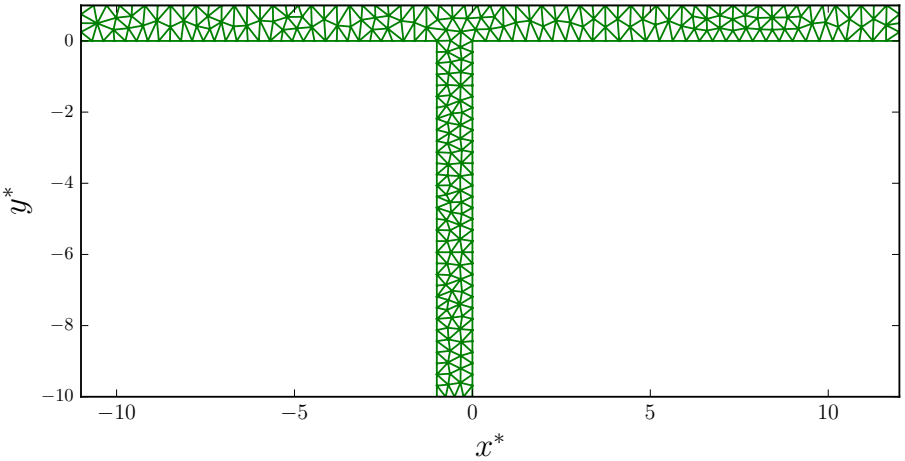
of the main and the lateral channels. The main channel length upstream of the lateral channel is lengthened to $12W = 10.97$ m and the lateral channel length is taken to be $10W = 9.14$ m. Since the water height is constant near the downstream boundary, the main channel is shortened to $10W = 9.14$ m downstream of the lateral channel. In Figure 3.11, we show how the domain is decomposed for the channel network model. Channels 1, 2 and 3 are treated as 1-D channels and are discretized along the centerline with a uniform $h = 0.25$ m. The junction is represented by a triangular mesh with $h_{max} = 0.25$ m. For comparison, we have chosen two different junction sizes. In the first case, the junction extends from $-5W$ to W in the horizontal direction and from $-W$ to W in the vertical direction. In the second case, the junction extends from $-2W$ to W in the horizontal direction and from $-W$ to W in the vertical direction. These junction meshes along with the fully 2-D mesh are shown in Figure 3.12. The mesh for the smaller junction shown in Figure 3.12a contains 68 triangular elements and that for the larger junction shown in Figure 3.12b contains 120 triangular elements. For the fully 2-D simulation, we represent the entire domain with a triangular mesh with $h_{max} = 0.25$ m. The fully 2-D mesh contains 490 triangular elements.



(a) Smaller junction



(b) Larger junction

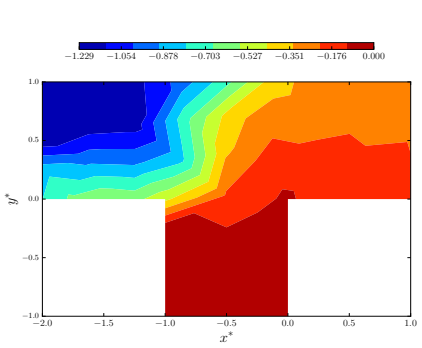


(c) A fully 2-D representation of the domain

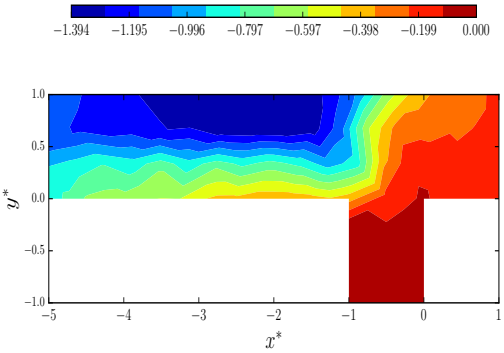
Figure 3.12: 2-D meshes for simulations of flow through a 90° junction

In Figure 3.13, we have plotted the normalized velocity in the x-direction obtained from our channel network model with the small junction (Figure 3.14a), the channel network model with the large junction (Figure 3.14b) and the fully 2-D model (Figure 3.15c). These results were obtained by running the simulation to time $t = 15000$ s, by which time, the system seems to have reached steady state. We have also shown the experimental data obtained

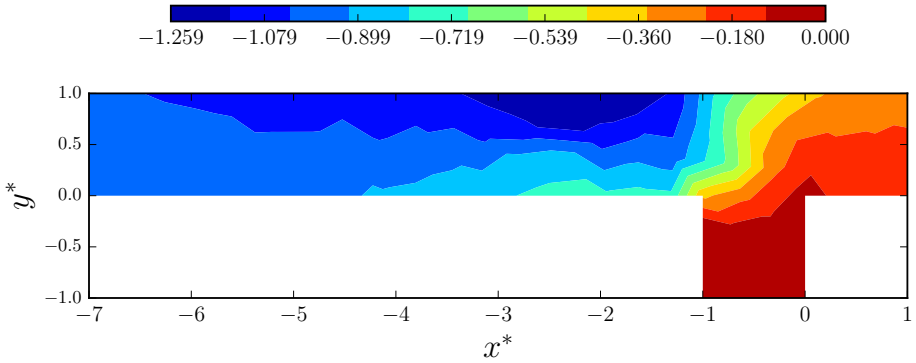
in [93]. The experiment was 3-D in nature, and thus the experimental data is reported at $z^* = 0.278$, where z is the vertical coordinate and z^* is the z-coordinate normalized by W. Since the velocity value we obtain from our simulations are the depth-averaged values, naturally, these are going to be different from the experimental data. The important aspect is the qualitative behavior of the solution. All three simulations reproduce the flow stagnation at the point $(x^* = 0, y^* = 0)$. The numerical simulations do not reproduce the re-circulation along the main channel wall closest to the lateral channel immediately downstream of the junction. However we have observed that with a finer $h = 0.1$, all three models reproduce this behavior as shown in Figure 3.14. These models do reproduce the contracted flow region, seen as the region of the highest velocity immediately downstream of the junction (or near in the case of the smaller junction) along the main channel wall opposite to the lateral channel.



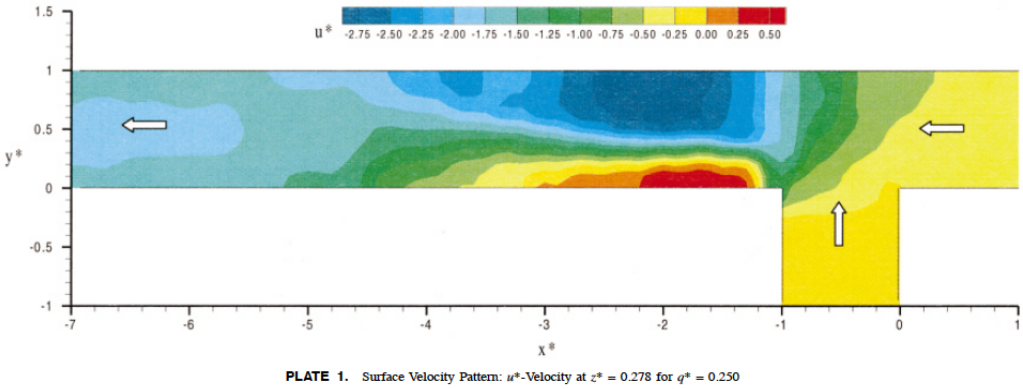
(a) A smaller 2-D junction



(b) A larger 2-D junction

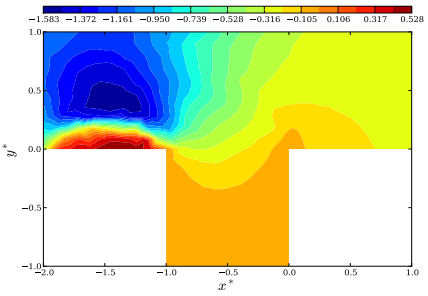


(c) Fully 2-D Simulation

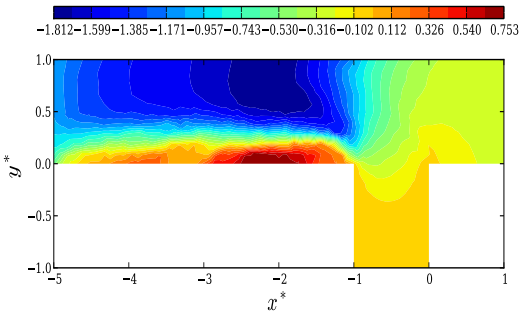


(d) Experimental data presented in [93]. Figure reused with permission from ASCE.

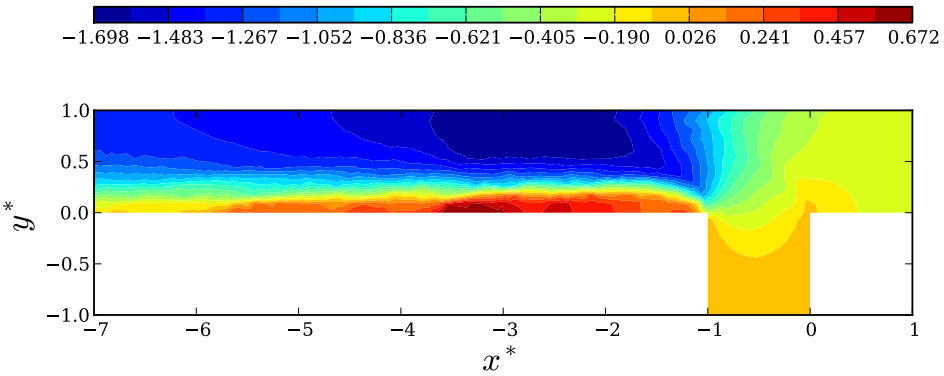
Figure 3.13: normalized velocity in the x-direction



(a) A smaller 2-D junction



(b) A larger 2-D junction



(c) Fully 2-D Simulation

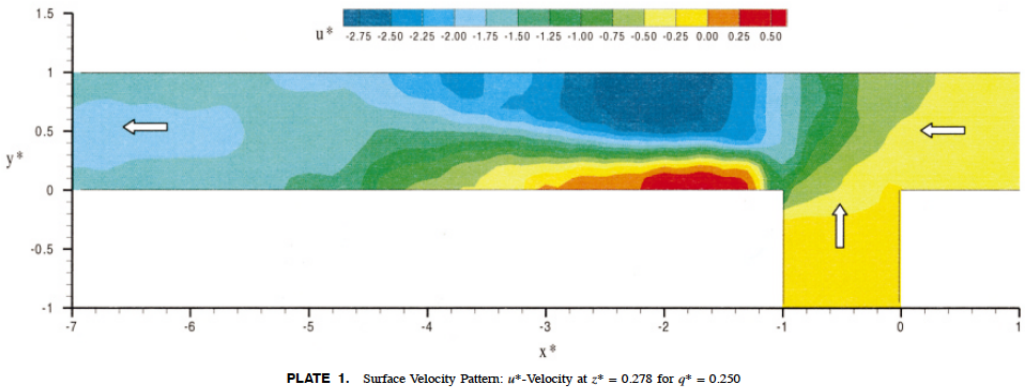
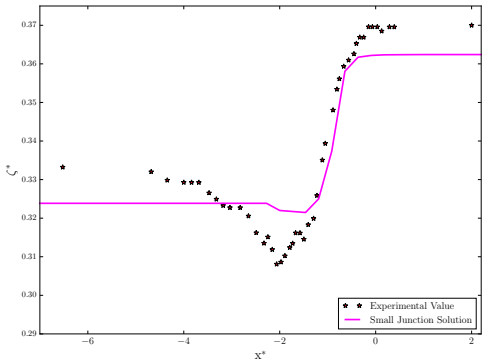


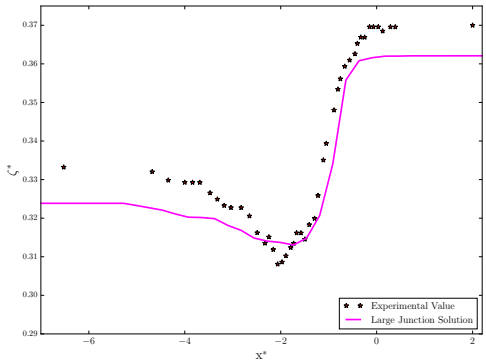
PLATE 1. Surface Velocity Pattern: u^*-Velocity at $z^* = 0.278$ for $q^* = 0.250$

(d) Experimental data presented in [93]. Figure reused with permission from ASCE.

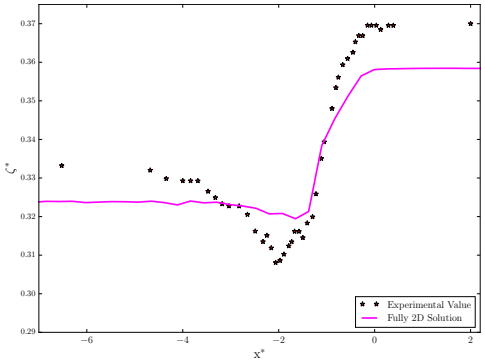
Figure 3.14: normalized velocity in the x-direction obtained using h=0.1



(a) Small junction simulation



(b) Large junction simulation



(c) Fully 2-D simulation

Figure 3.15: Normalized water surface height

In Figure 3.15, we have plotted the normalized water surface height obtained from the three different simulations against experimental data presented in [93]. All of the simulations reproduce the dip in the water height near $x^* = -2$. This dip exists because of the zone of separation created by the lateral flow coming into the main channel from the lateral channel near $x^* = -1$. The errors we obtained are not negligible in any of the simulations. However, better results can be obtained with finer discretizations.

Model	Raw l2-error	Normalized l2-error	Maximum Error	Percentage Error
Small Junction	0.047	0.021	0.014	0.047
Large Junction	0.044	0.019	0.012	0.035
Fully 2-D Model	0.053	0.024	0.013	0.041

Table 3.1: Error in water surface height

In Table 3.1, we present the error for all the different simulations. The raw l2-error is calculated by taking the l2-norm of the difference in the solution at the points reported in the experimental data. The normalized l2-error is the l2-error normalized by the l2-norm of the experimental solution. The maximum error is the maximum value of the difference between the experimental data and the numerical solution at those points. The percentage error is the maximum error normalized by the experimental solution at the point where the error is the highest.

We see that the errors in our channel network model with the small junction are similar to the fully 2-D simulation. The large junction model produces the best result. The runtime is significantly lower for both of the channel network model simulations than for the fully 2-D simulation. This may seem like an argument in favor of using fully 1-D models, however, a fully 1-D model will not produce an accurate result either. Observe in Figure 3.15a that the solution obtained with small junction decomposition in the region $x^* = -7$ to $x^* = -2$ is much flatter compared to the experimental results and the results in Figure 3.15b. This is because, in the small junction decomposition,

this region is represented by a 1-D channel, however it is represented as a 2-D region in the large junction decomposition. Thus, it is evident that the region needs to be represented as a 2-D region to capture the zone of separation.

Chapter 4

Overland Flow

In this chapter, we discuss the simulation of rainfall runoff in overland flow regions. Recall that overland flow regions are the dry land regions that receive rainfall. The rainwater that has not infiltrated the land flows down the land surface and eventually gets collected in channels. Here, we first discuss the solution of the kinematic wave equation. Then, we discuss various models that are used to model infiltration. Finally, we end the chapter with a discussion of how we use the kinematic wave equation to model runoff in a two-dimensional watershed.

4.1 Solution of the Kinematic Wave Equation

We solve the kinematic wave equation given in (2.8) using the second order Runge–Kutta method described in Section 2.3. We use the same technique that we use to solve the channel equations and that were described in Section 2.3.2 to discretize the kinematic wave equation. Because of the hyperbolic nature of the kinematic wave equation, information travels from the upwind direction. Thus, we use the upwind flux as the numerical flux in the discontinuous Galerkin discretization given in (2.12). The upwind flux is given

by:

$$\hat{F} = F_L \tag{4.1}$$

where F_L is the value of the flux on the upstream side.

We test our numerical scheme on two synthetic examples described by Kazezyilmaz-Alhan et al. [49].

I. Storm of constant rainfall intensity

Here, we test our code on a problem for which the analytical solution of the kinematic wave equation is available. One such problem is the case of constant rainfall on an impervious surface with initial condition $H = 0$ at $t = 0$ and boundary condition $H = 0$.

Consider the situation where a 600 ft long impervious parking lot receives rainfall at an intensity, $i_* = 2$ in/hr for 30 minutes. The slope of the parking lot is 0.0016, which is close to the lower limit of the range of slopes over which the kinematic assumption is valid as described in Section 2.1. Manning's roughness coefficient (n) for the parking lot is taken to be 0.01628 s/$\sqrt[3]{\text{ft}}$. We are interested in the outflow rate at the end of the parking lot for 60 minutes. Kazezyilmaz-Alhan et al. [49] derived the following analytical solution to the kinematic wave equation under the conditions described above.

$$\begin{aligned}
&q = \frac{1}{n}\sqrt{S_0}(i_* t)^{\frac{5}{3}} && \text{for } 0 < t < 1494 \text{ s} \\
&q = 1.67 \text{ ft}^2/\text{min} && \text{for } 1494 \text{ s} < t < 1800 \text{ s} \\
&t - 1800 = (600 - 21600q)/(2.82q^{\frac{2}{5}}) && \text{for } t > 1800 \text{ s}
\end{aligned} \tag{4.2}$$

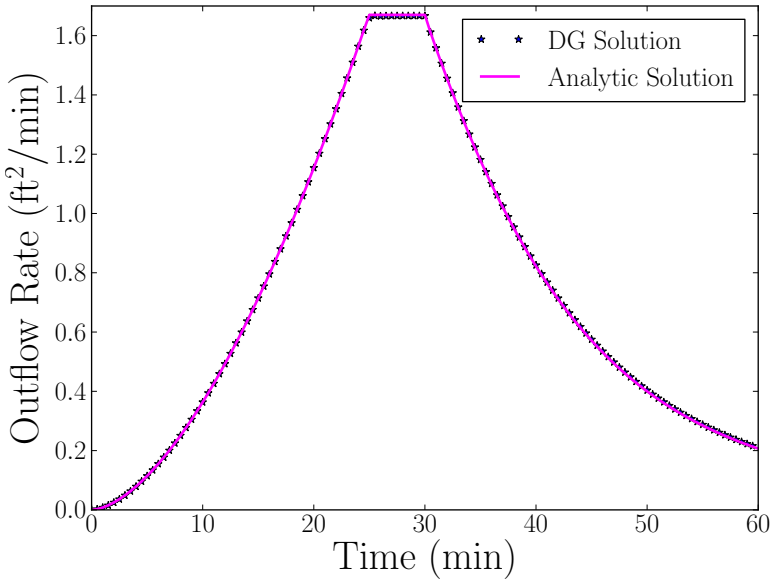


Figure 4.1: Numerical solution of the storm with constant rainfall intensity plotted against the analytical solution

We solve the kinematic wave equation given in (2.8) for the water height, H. Then, we use Manning's relation: $u = \frac{1}{n}\sqrt{S_0}H^{\frac{2}{3}}$ to evaluate u and calculate the outflow rate $q = Hu$ at the end of the parking lot. In Figure 4.1, we have plotted the solution obtained from our numerical scheme referred to as the "DG Solution" in the plot against the analytical solution. The DG solution was obtained using 400 elements to discretize the parking lot using first order polynomial approximation. We see that our analytical solution matches the hydrograph very well, both while it is raining and after the rain has stopped.

II. Storm of varying rainfall intensity

Here, we test our code for a problem to which no analytical solution is available, but will give us some insight on whether our numerical scheme can capture the nonlinearity of the problem. Instead of using Manning's equation for the relationship between the velocity and water height, we use Chezy's equation. Using Chezy's formula, $u = C\sqrt{HS_0}$, where C is Chezy's friction coefficient given in $\sqrt{\text{m}}/\text{s}$, we obtain the following form of the kinematic wave equation:

$$\frac{\partial H}{\partial t} + \frac{\partial}{\partial x}\left(C\sqrt{S_0}H^{\frac{3}{2}}\right) = q_L \tag{4.3}$$

The problem we want to model with this equation is the case of a 100 ft long parking lot receiving rainfall at an intensity of 2 in/hr for the first 3 minutes, after which the rainfall intensity increases to 4 in/hr for the second 3 minutes before coming to a stop. The slope of the parking lot is 0.005. Chezy's friction coefficient for the parking lot is 71 $\sqrt{\text{m}}/\text{s}$. We are interested in the outflow rate at the end of the parking lot for 9 minutes. Since this storm brings rainfall of varying intensity, an analytical solution cannot be derived. However, a "semi-analytical" solution can be derived by calculating the analytical solutions for the two rainfall intensities separately as two different storms. Kazezyilmaz-Alhan et al. [49] derived the following analytical solutions for the two rainfall intensities:

For $i_* = 2$ in/hr:

$$H = i_* t \qquad \text{for } 0 < t < 180 \text{ s}$$

$$H = 0.00828 \text{ ft} \qquad \text{for } 180 \text{ s} < t < 207 \text{ s}$$

$$t - 180 = 13.3/\sqrt{H} - H/0.000069 \qquad \text{for } t > 207\text{s}$$

For $i_* = 4$ in/hr:

$$H = i_* t \qquad \text{for } 180 \text{ s} < t < 344 \text{ s}$$

$$H = 0.015 \text{ ft} \qquad \text{for } 344 \text{ s} < t < 360 \text{ s}$$

$$t - 180 = 13.3/\sqrt{H} - H/0.000139 \qquad \text{for } t > 360 \text{ s}$$

The superposition of these two solutions is the "semi-analytical" solution.

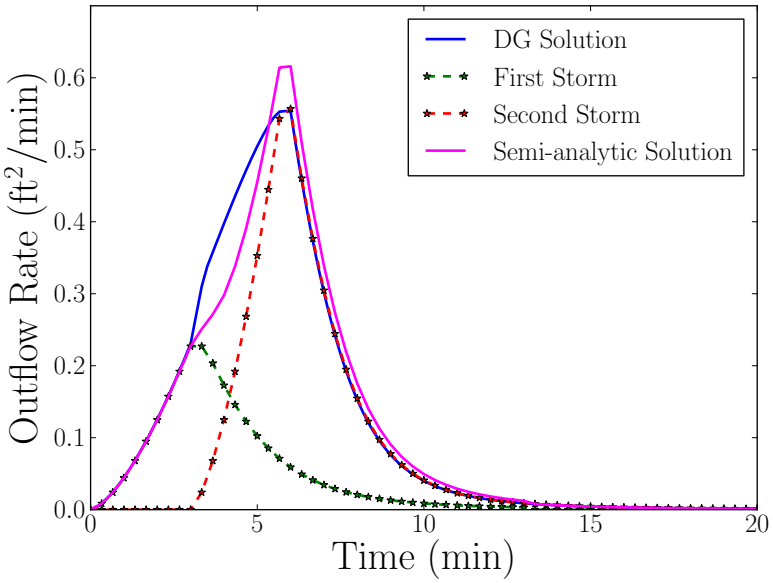


Figure 4.2: Numerical solution of the storm with variable rainfall intensity plotted against the "semi-analytical" solution

In Figure 4.2, we have plotted the solution obtained from our numerical scheme, referred to as "DG Solution" in the plot. The "First Storm" solution is the solution for the problem with the constant rainfall intensity of 2 in/hr for the first 3 minutes. The "Second Storm" solution is the solution for the problem with the constant rainfall intensity of 4 in/hr from minutes 3 to 6. The "Semi-analytic Solution" is the superposition of the first storm and the second storm solutions. As expected, we see that the DG Solution starts departing from the semi-analytic solution when the flood hydrograph from the first storm starts descending and the second storm starts developing. This is because a change in rainfall intensity is not really a different storm that is independent of the first one. However, the DG solution does match the peak of the first storm as well as that of the second storm.

Thus, we have established confidence in our numerical scheme.

4.2 Infiltration Model

When a surface receives rainfall, a very important factor that determines how much water flows down the surface as runoff is the infiltration capacity of the surface. Infiltration is the process of water penetrating from the ground surface into the soil. The infiltration rate $f(t)$ with dimension L/T is determined by a number of factors including the conditions of the soil surface, its vegetative cover, the porosity of the soil, the hydraulic conductivity of the soil and the current moisture content of the soil.

Infiltration is a complex process, and the equations that aim to describe

this process should take the following factors into account: If water is ponded on a surface, infiltration through the surface should occur at the *potential infiltration rate*, i.e. the maximum infiltration capacity of the surface. If the rate of supply of water at the surface is less than the potential infiltration rate then the actual infiltration rate should be less than the potential infiltration rate. Below, we briefly describe the three methods used to describe infiltration. All of these methods assume that a small amount of water has ponded over the surface so that infiltration can occur at the maximum infiltration capacity. Please refer to [15] for additional details.

I. Horton's equation

Horton's equation is given by:

$$f(t) = f_c + (f_0 - f_c)e^{-kt} \tag{4.4}$$

where k is a decay constant with dimension T^{-1}. The equation above expresses that infiltration begins at some rate f_0 and decreases exponentially to a constant rate f_c.

II. Phillip's equation

Phillip's infiltration equation is given by:

$$f(t) = \frac{1}{2}S/\sqrt{t} + K \tag{4.5}$$

where S is the sorptivity of the soil and has dimension $L/\sqrt{T}$ and K is the hydraulic conductivity with dimension L/T. Sorptivity is a measure of the

capacity of the soil to absorb or desorb liquid by capillarity. The equation above expresses that, initially, infiltration is a function of the soil suction head as well as the gravity head. As $t \to \infty$, $f(t) \to K$, which means that as time goes on, infiltration is driven only by the gravity head.

III. Green–Ampt equation

The cumulative infiltration, F, is defined as the accumulated depth of water infiltrated during a given time period and is equal to the integral of the infiltration rate over that period. The Green–Ampt equation for cumulative infiltration is as follows:

$$F(t) = Kt + \psi\ \Delta\theta\ ln\left(1 + \frac{F(t)}{\psi\ \Delta\theta}\right) \tag{4.6}$$

In the equation above, K is the hydraulic conductivity, θ is the soil moisture content and ψ is the soil suction head and has dimension L. The soil moisture content is defined as:

$$\theta = \frac{\text{volume of water}}{\text{total volume}} \tag{4.7}$$

$\Delta\theta$ can be written in terms of the effective porosity, θ_e and the effective saturation s_e as follows:

$$\Delta\theta = (1 - s_e)\ \theta_e \tag{4.8}$$

where θ_e is the effective porosity and s_e is the effective saturation. Effective saturation is given by:

$$s_e = \frac{\theta - \theta_r}{\eta - \theta_r} \tag{4.9}$$

where η is the porosity and θ_r is the residual moisture content of the soil after it has been thoroughly drained. Thus, we see that effective saturation is the ratio of the available moisture to the maximum possible available moisture content $\eta - \theta_r$. Assuming that $\theta_r \leq \theta \leq \eta$, the value of s_e will be in the range $[0, 1.0]$.

The values of the various parameters needed for the Green–Ampt equations like the porosity (η), the effective porosity (θ_e), the suction head (ψ) and the hydraulic conductivity (K) have been tabulated for different kinds of soils (See [15]). After plugging in these values, equation (4.6) can be solved for $F(t)$ using an iterative method like successive substitution or Newton's iterative method. Once the cumulative infiltration rate has been calculated, the infiltration rate can be calculated using the following formula:

$$f(t) = K\left(\frac{\psi\ \Delta\theta}{F(t)} + 1\right) \tag{4.10}$$

4.2.1 Rainfall on a Pervious Surface

In order to simulate flow induced by rainfall on a pervious surface, we use the Green-Ampt infiltration model in the kinematic wave equation. At every time step, we solve (4.6) iteratively using Newton's method. The source term for the kinematic wave equation is $q_L = R - I$, where I is given by the Green–Ampt infiltration equation (4.10).

We consider a parking lot and a storm similar to the one in Section 4.1 where we tested our numerical scheme on a storm of constant rainfall intensity.

The parking lot is 600 ft long, but now it is made of sandy clay loam soil so that the soil is pervious and rainwater can infiltrate through the soil. For the purposes of comparison, the slope and the Manning's friction coefficient of the parking lot are kept the same as in Section 4.1. In order to use the Green-Ampt infiltration model, we need the values for certain parameters that appear in the model. For a sandy clay loam soil, the following values are commonly used for the Green–Ampt parameters:

$$K = 0.15 \text{ cm/hr}, \qquad \theta_e = 0.330, \qquad \psi = 21.85 \text{ cm}.$$

Rainfall continues on the parking lot for 30 minutes at an intensity of 2 in/hr. We assume that there has already been some ponding on the parking lot, so that the Green–Ampt equation applies and that the initial saturation of the soil is 0, i.e. $s_e = 0$. We measure the outflow rate at the end of the parking lot for an hour.

In Figure 4.3, we have plotted the solution for the sandy clay loam surface along with the solution when the surface is impervious, labeled "concrete". We see that the water starts draining out of the concrete parking lot sooner than out of the sandy clay loam parking lot. Initially, all the water infiltrates into the parking lot, hence we observe that for about the first 5 minutes, there is zero outflow at the end of the parking lot. As water starts flowing out of the parking lot, we see that the outflow rate at the end of the parking lot is lower than that for the concrete parking lot. This is to be expected, because the completely unsaturated sandy clay loam parking lot is able to absorb some of

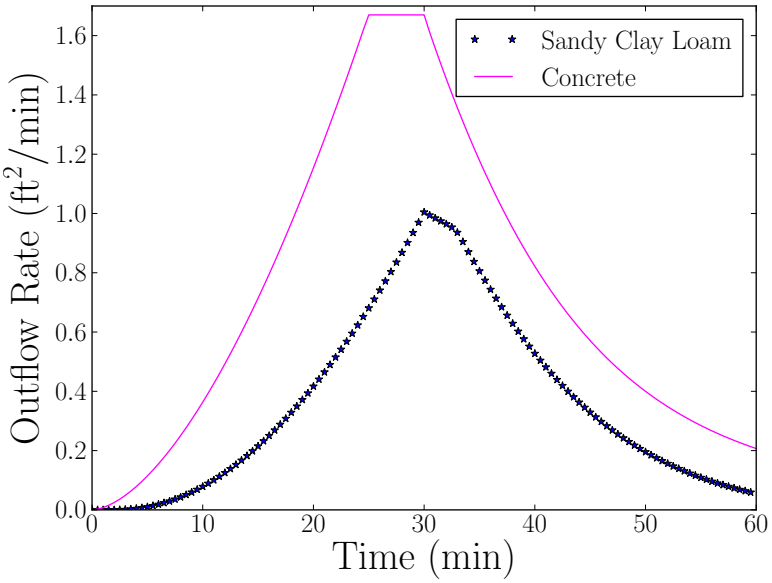


Figure 4.3: Numerical solution of the storm that brings rainfall of constant with and without infiltration

the rainwater before it flows down the surface. Thus, we see that the chances that a channel that receives rainfall runoff will flood is decreased if the water can infiltrate the surface receiving rainfall.

In the next section, we discuss how we can use the 1-D kinematic equation for modeling 2-D floodplains.

4.3 Kinematic Wave Modeling of Overland Flow Region

The first step in hydrologic modeling of a watershed is the creation of a drainage network either from a raster data structure (also called grids) or a TIN (Triangular Irregular Network) data structure representation of a watershed. The elevation values assigned to the triangular or the grid cells can be used

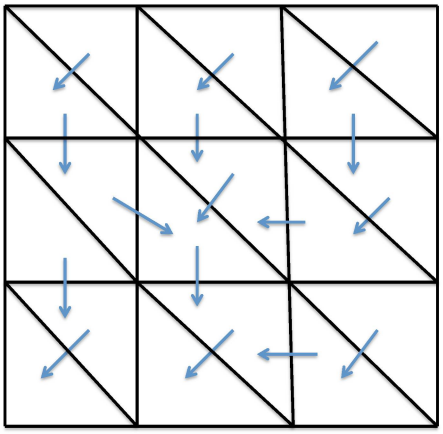


Figure 4.4: An illustration of 1-D representation of a 2-D domain

to calculate the flow direction for each cell. Various algorithms have been proposed to assign the flow direction for each cell in order to automatically detect channel networks and watershed boundaries (e.g. [67, 71, 89]). The D8 method introduced by O'Callaghan and Mark [71] assigns flow from each pixel or a grid cell to one of its eight neighbors, either adjacent or diagonal, in the direction with the steepest downward slope. We adapt this method to a mesh discretized with triangular elements by assigning each cell three possible drainage directions – one through each edge. Once the flow direction has been assigned to each cell discretizing the overland flow region, we have a complete representation of the 2-D region with 1-D segments.

In Figure 4.4, we illustrate what such a representation looks like. The

arrows represent the direction in which flow from a certain element will travel. These arrows can also be thought of as the 1-D elements that discretize the 2-D domain. Vieux [92] describes this approach on a grid in a finite element method framework for distributed modeling of watershed hydrology. We use a discontinuous Galerkin finite element method in a similar manner. Similar to the framework Vieux describes, we prescribe the surface elevation and the surface roughness values at the nodes. The nodal values will be linearly interpolated to the whole domain resulting in a constant slope over one element. The approach of prescribing these values at the nodes as opposed to on an element alleviates the need of breaking a watershed into equivalent conceptual cascades, planes or subareas [92].

In order for the 1-D equations to account for the entire area of the 2-D region, we need to associate a width with each 1-D element similarly to the 1-D Saint Venant equations. This can be achieved by defining a quantity called "equivalent width," denoted as w_{eq}. It is defined as follows:

$$w_{eq} \equiv \frac{A_{OF}}{L_{kin}} \tag{4.11}$$

where A_{OF} is the area of the overland flow region and L_{kin} is the total sum of the length of the overland flow elements. Now, a flow area for the overland flow elements similar to the cross-sectional area of flow for channel elements can be defined as: $A = w_{eq}H$. Using this definition, the kinematic wave equation can be rewritten as follows:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial s} = w_{eq} q_L \tag{4.12}$$

Q is now the volumetric discharge. Noting that w_{eq} is constant over the entire overland flow region, the equation above can be reduced down to the kinematic wave equation (2.8) again. However, we will be using the concept of the equivalent width later when letting flow from the overland flow region drain into the channels.

4.3.1 Discontinuous Galerkin finite element method simulation of overland flow region

In this section, we want to test how well the 2-D scheme for the kinematic wave equation captures the kinematic flow in a 2-D region. We discretize the kinematic wave equation using the RKDG method previously described. Again, we use the upwind flux given in (4.1) as the numerical flux. A complication arises at the "junction" of three kinematic elements, i.e. when two upstream kinematic elements (call them el_1 and el_2) direct flow to the same downstream element (call it el_3). At the downstream ends of the two upstream kinematic elements, the upwind flux is prescribed as usual, i.e.

$$\hat{F}_{el_1} = F_{el_1} \text{ and } \hat{F}_{el_2} = F_{el_2} \tag{4.13}$$

The upwind flux at the upstream end of the downstream kinematic element is taken to be

$$\hat{F}_{el_3} = F_{el_1} + F_{el_2} \tag{4.14}$$

where F_{el_1} and F_{el_2} are the values of the flux at the two upstream elements respectively. This approach ensures that mass is conserved at these junctions.

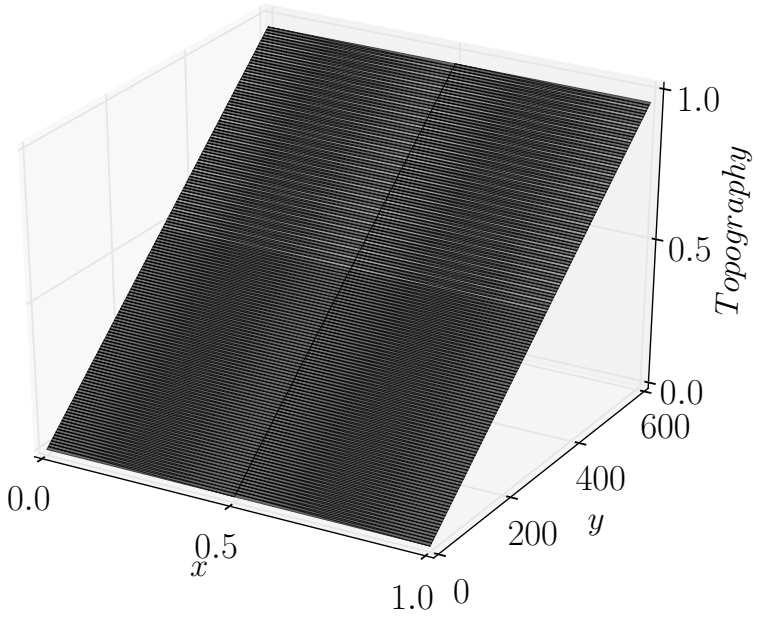


Figure 4.5: 2-D discretization of the parking lot for the constant rainfall intensity test case

The test case we use is one of the test cases we used for testing the numerical scheme for the 1-D kinematic wave equation. We have a 600 ft long parking lot that receives rainfall at a constant intensity for half an hour, and we observe the outflow rate at the end of the parking lot for an hour. See Section 4.1 for details. Now, we represent the parking lot as a 2-D domain having unit width. We discretize this parking lot using triangular elements as shown in Figure 4.5, using 2 elements in the x-direction and the same number of elements (400) in the y-direction as we did for the simulation in Section 4.1.

The three edges of a triangular element are the three possible flow directions. We investigated two different ways of assigning the flow direction to each element:

Approach I:

- Calculate the elevation at the midpoint of each edge of the triangle.
- Assign the edge with the lowest elevation as the flow edge.
- If two edges have the same elevation, pick an element at random.

Approach II:

- Linearly interpolate the nodal elevation representing elevation as a continuous surface
- Calculate the gradient of the surface elevation at the barycenter of each element
- Calculate the flow vector corresponding to each edge. The three flow vectors are defined as the three vectors pointing from the barycenter of the element to the barycenters of each of the three neighboring elements.
- Pick the edge that borders the element resulting in the flow vector "closest" to the gradient as the flow edge. The "closeness" is measured by the dot product of the gradient vector with the flow vector.

In Figure 4.6, we have plotted the total outflow rate achieved at the end of the parking lot. The total was calculated as the sum of the discharges at the two elements that discretize the outflow boundary of the parking lot. The first approach to calculating flow paths results in a lower outflow rate than the analytical solution. This is not surprising, because this approach might calculate flow paths that might not be entirely physically accurate. The elevation of the parking lot in our simulation increases linearly in the y-direction and is constant in the x-direction. Consider now a triangle with coordinates $(0.5, 0)$, $(0.5, 1.5)$ and $(1.0, 1.5)$. The elevation values at the midpoint of the edge connecting vertices $(0.5, 0)$ and $(0.5, 1.5)$ and at the midpoint of the edge connecting vertices $(0.5, 0)$ and $(1.0, 1.5)$, are equal. Since the algorithm has no preference when the midpoint of two sides of an element have the same elevation, flow might be routed towards the edge connecting vertices $(0.5, 0)$ and $(0.5, 1.5)$ when physically, it should be routed towards the edge connecting vertices $(0.5, 0)$ and $(1.0, 1.5)$. When this happens, the flow path is artificially lengthened. This results in the flow being subjected to a longer flow path, and in essence a higher resistance. As a result, the outflow rates are lower as observed in Figure 4.6.

The second approach works well for the simulation domain we chose because our parking lot has only one direction of steepest descent. In cases where there might be two directions of steepest descent, the gradient algorithm might have difficulties choosing the right flow direction as well. There have been works that handle situations like this when there can be more than one

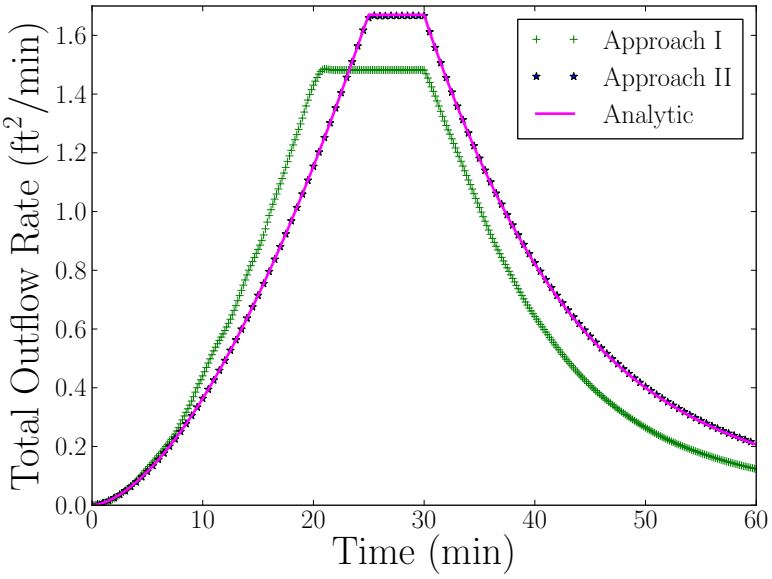


Figure 4.6: Total discharge rate at the end of the parking lot using Approach I and Approach II

possible receiving element (See for e.g. [48]). In a real watershed, numerous other difficulties can arise. For example, an element might be a sink, i.e. all the three edges of the element might have the same elevation values. This can happen either because the watershed actually has some depressions or as an artifact of discretization and having information available only at the vertices. This will cause difficulties while using the kinematic wave equation, because a nonzero slope is required to drive flow. Another unreal attribute of flow is introduced when flow paths are calculated by constraining the flow to one of three possible directions in the case of triangular discretization and one of eight possible directions in the case of a grid. Various works have tried to address this shortcoming as well, either by proposing multiple flow directions from a cell, or by specifying flow direction continuously as an angle between 0

and 2π. The reader is referred to [33, 59, 73, 89] for details.

Chapter 5

Coupling of Sub-Domains

In this chapter, we begin to view a coastal watershed as a single domain comprising all the different sub-domains. Here, we discuss how different sub-domains are coupled with each other to allow proper routing of flow from one sub-domain to another. We begin by describing how an overland flow region is coupled with a network of channels. Then, we describe how the inland regions of a watershed are coupled with the shallow water region near the coast through a channel. Finally, we describe the conditions under which a channel is considered to have flooded. We end the chapter with a discussion of how a flooded channel is coupled with floodplains.

5.1 Coupling of Overland Flow Region with Channel Networks

In a natural watershed, the rainwater that travels down a hill slope eventually drains down to a river or drainage channel. In our modeling framework, we treat channels and hillslopes as separate domains; we use different equations to model flow in these sub-domains. Therefore, we need to specify a way to couple the two sub-domains with each other. We need to ensure that when we couple these two sub-domains with each other, all the water that

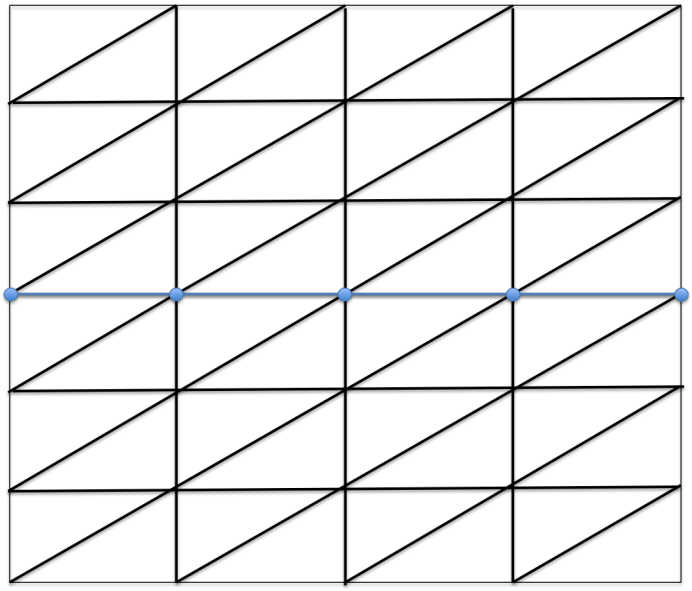


Figure 5.1: Schematic of a mesh of a watershed consisting of hillslopes on either side of the channel shown in blue

flows out of the hillslopes does indeed collect in the channel, i.e. the coupling needs to be mass conservative.

In our current approach, only the channels in a channel network are coupled to hillslopes. The junctions do not receive flow from hillslopes. Therefore, here we focus on the coupling of a channel with overland flow regions. Consider a watershed with a channel that runs through its center. We have shown the mesh for this watershed in Figure 5.1. The blue circles in the mesh represent the vertices of channel elements; the blue line segments represent channel elements. There is an overland flow region on either side of this channel. We have shown how we decompose this domain in our modeling framework in Figure 5.2. The blue line in Figure 5.1 represents the channel shown in blue in

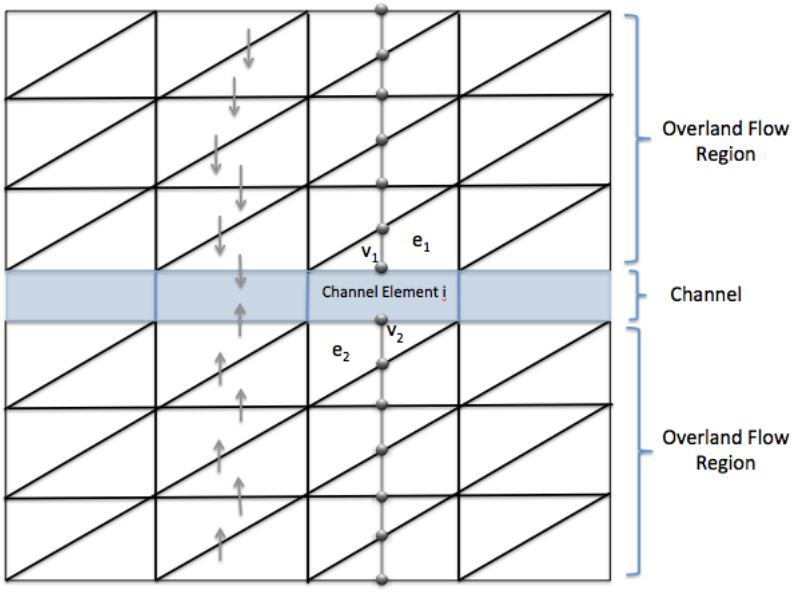


Figure 5.2: Domain decomposition of the watershed shown in Figure 5.1. The channel is shown as having a certain width.

Figure 5.2. The physical channel is not 1-D. It has an associated width. The grey arrows represent the flow directions, which are used to create the 1-D discretization of the 2-D domain. The grey line segments represent the 1-D kinematic elements on which the kinematic wave equation is solved. The grey circles represent the vertices of the 1-D kinematic elements that now discretize the 2-D hillslopes.

Recall the continuity equation for channel flow:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial s} = q_L \tag{5.1}$$

Using this equation, we allow channels to receive flow from the overland flow region as lateral flow, q_L. Recall that in a DG framework, the equation above is solved in each channel element. Now consider the channel element i in

Figure 5.2 that is connected to overland flow elements e_1 and e_2. Effectively, the channel element i is connected to the vertices v_1 and v_2 of the 1-D kinematic elements. On the channel element i, we prescribe q_L in the following manner:

$$q_{L,i} = \sum_{e=1}^{2} \frac{w_{eq}}{L_i} (Hu)_{v_e} \tag{5.2}$$

L_i is the length of the channel element i, w_{eq} is the equivalent width defined in (4.11). $(Hu)_{v_e}$ is the discharge rate at vertex v_e. In the discrete scheme, we update $\tilde{\mathbf{A}}_i$, the vector containing the values of the wet cross-sectional area of flow at the nodes on channel element i, in the following manner at the $(N+1)^{st}$ time step:

$$\tilde{\mathbf{A}}_i^{N+1} = \tilde{\mathbf{A}_\mathbf{i}}^{N} + \Delta t\, \mathbf{M_i^{-1}} \mathbf{S_i^T} \tilde{\mathbf{Q}}_i^N - \mathbf{M_i^{-1}} \mathbf{L_i} \tilde{\tilde{\mathbf{Q}}}_i^N + \tilde{\mathbf{q}}_\mathbf{L}^N \tag{5.3}$$

where, $\tilde{\mathbf{q}}_\mathbf{L}^N = \begin{pmatrix} q_{L,i}^N \\ \vdots \\ q_{L,i}^N \end{pmatrix}$.

Notice that since all the values in the vector $\mathbf{q_L}$ of nodal values of the function q_L are the same, q_L is a constant function over a channel element, and is piece-wise constant over the entire channel. Notice also that this is a one way coupling. The overland flow region has no mechanism to receive any flow information from the channels. This is because the kinematic wave equation used in the overland flow region does not allow information to travel up the domain from the downstream direction. It should be noted that while the scope of this coupling might be limited to a one-way propagation of water mass, this coupling does conserve the water mass. We will show this both analytically as well as via a numerical simulation in the subsequent sections.

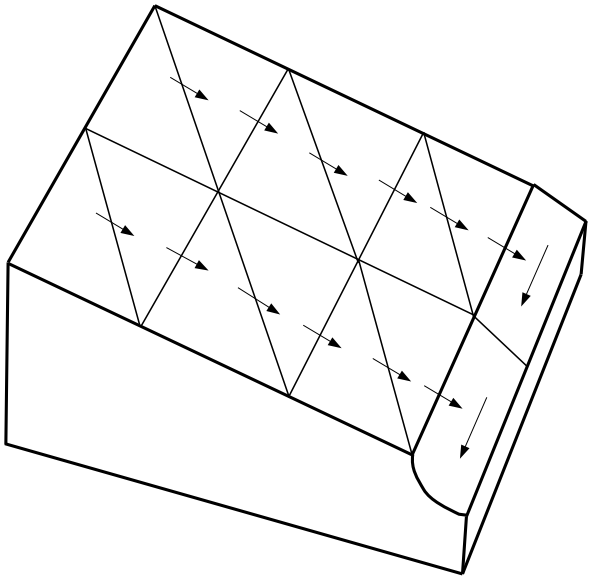


Figure 5.3: Schematic of overland flow draining into a channel from the parking lot

5.1.1 Numerical Test of Mass Conservation

Consider again the parking lot we simulated in Section 4.3.1. Rain falls on the parking lot for 30 minutes at an intensity of 2 in/hr. We add a channel at the end of the parking lot as shown in Figure 5.3. The channel is flat. We discretize it such that the edges of the triangular overland flow element connected to the channel are the channel elements that discretize the channel. Initially, it is dry. We allow the water flowing out of the parking lot to drain into the channel. We enforce a no-flow boundary condition at both ends of the channel and measure the volume of the water collected in the channel over time.

In Figure 5.4, we have plotted the volume of the water collecting in the channel as a function of time. We used both approaches described in Sec-

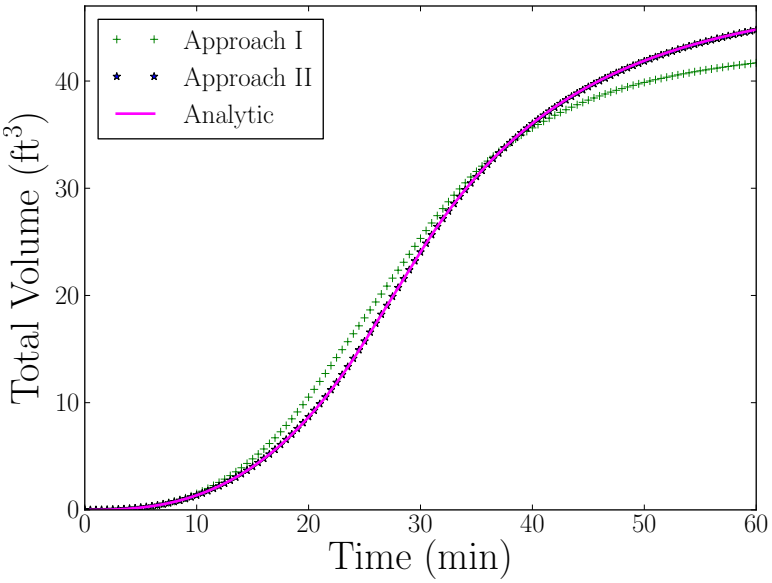


Figure 5.4: Total water collected in the channel

tion 4.3.1 to calculate the flow paths for the parking lot. The "analytical" solution plotted in the figure was obtained by integrating in time the analytical solution for the outflow rate given in (4.2). The integration was done numerically using a simple midpoint rule. The total volume collected in the channel was calculated by integrating the wet cross-sectional area (A) of the channel in space. We see that, as expected, the water collected in the channel using approach I does not match the analytical solution. While initially, more water than expected is collected in the channel, at the end of an hour, the total water in the channel is less than expected. This is unsurprising because as we saw in Figure 4.6, the outflow rate using this approach was initially higher, but never reached the peak of the analytical outflow rate. The second approach does exactly match the analytical solution. The total volume of water (V)

that the parking lot received from the half hour of rainfall can be calculated as follows:

$$V = A_p * i_* * t$$

where A_p is the area of the parking lot, i_* is the rainfall intensity and t is the duration of the rainfall. For our test case, $V = 50$ ft^3. At the end of an hour, we have collected about 44 ft^3 of water in the channel. Notice that the outflow rate shown in Figure 4.6 at the end of an hour is not zero yet. Thus water is still draining out of the parking lot. As we let the simulation run longer, we will eventually have collected all of the rainwater in the channel.

5.1.2 Theoretical Analysis of Mass Conservation

Now, we will show analytically that the DG scheme for channel flow and overland flow along with the coupling condition given in (5.2) results in a mass conservative scheme for the coupled system.

Consider a domain with one channel and an overland flow region as shown in Figure 5.3. Let A denote the numerical approximation of the cross-sectional area of flow in channels. Let Q denote the numerical approximation of the volumetric discharge in channels. Let H denote the numerical approximation of the water height in the overland flow region. Let u denote the numerical approximation of the velocity in the overland flow region.

Recall that A satisfies the following weak form of the continuity equa-

tion:

$$\left\langle \frac{\partial A}{\partial t}, \psi \right\rangle_{s_i} = \hat{Q}\psi|_{\mathbf{s}_{i-\frac{1}{2}}} - \hat{Q}\psi|_{\mathbf{s}_{i+\frac{1}{2}}} + \left\langle Q, \frac{\partial \psi}{\partial s} \right\rangle_{s_i} + \langle q_L, \psi \rangle \qquad \forall \psi \in V \quad (5.4)$$

where s_i is a channel element that extends from vertex $\mathbf{s}_{i-\frac{1}{2}}$ to vertex $\mathbf{s}_{i+\frac{1}{2}}$, for $i = 1 \ldots n$, where n is the number of elements used to discretize the channel. Replacing ψ by the constant polynomial $1 \in V$ in (5.4), we obtain:

$$\int_{s_i} \frac{\partial A}{\partial t} \, \mathbf{ds} = \hat{Q}|_{s_{i-\frac{1}{2}}} - \hat{Q}|_{s_{i+\frac{1}{2}}} + \int_{s_i} q_L \, \mathbf{ds} \quad (5.5)$$

Summing up the above equation over all elements in the 1-D channel and recalling that the numerical flux, $\hat{Q}$, is continuous at the common vertex shared by two 1-D elements, we obtain:

$$\frac{d}{dt} \sum_i \int_{s_i} A \, \mathbf{ds} = Q_{in} - Q_{out} + \sum_i \int_{s_i} q_L \, \mathbf{ds} \quad (5.6)$$

where Q_{in} and Q_{out} are the discharges prescribed at the inflow and the outflow boundaries respectively. For this analysis, we will assume that there is no water flowing in or out of the channel, i.e. $Q_{in} = Q_{out} = 0$. Then the equation above becomes:

$$\frac{d}{dt} \sum_i M_{ch,i} = \sum_i \int_{s_i} q_L \, \mathbf{ds} \quad (5.7)$$

where $M_{ch,i}$ is the water mass in channel element s_i. Notice that in the equation above, we have made the substitution, $M_{ch,i} \equiv \int_{s_i} A \, \mathbf{ds}$.

Similarly, the weak form of the kinematic wave equation adopted for a two dimensional region can be written as:

$$\begin{aligned} \left\langle \frac{\partial (w_{eq} H)}{\partial t}, \phi \right\rangle_{k_j} &= w_{eq} \widehat{Hu\phi}|_{\mathbf{k}_{j-\frac{1}{2}}} - w_{eq} \widehat{Hu\phi}|_{\mathbf{k}_{j+\frac{1}{2}}} + \langle w_{eq} Hu, \phi \rangle_{k_j} \\ &+ < w_{eq} R, \phi >_{k_j} - < w_{eq} I, \phi >_{k_j} \end{aligned} \quad (5.8)$$

where k_j is a kinematic element that extends from vertex $\mathbf{k}_{j-\frac{1}{2}}$ to $\mathbf{k}_{j+\frac{1}{2}}$, for $j = 1 \ldots K$, where K is the number of 1-D kinematic elements that discretize the 2-D overland flow region. By replacing ϕ by the constant polynomial $1 \in V$ in (5.8) and adding the equation over all elements that discretize the 2-D overland flow region, we obtain the following equation:

$$\begin{aligned} \frac{d}{dt} \sum_j \int_{k_j} w_{eq} H \ \mathbf{ds} = & \sum_b w_{eq} (Hu)_b - \sum_e w_{eq} (Hu)_e \\ & + \sum_J w_{eq} \left((\widehat{Hu})_{el_3} - (\widehat{Hu})_{el_1} - (\widehat{Hu})_{el_2} \right) \\ & + \sum_j \int_{k_j} w_{eq} R \ \mathbf{ds} - \sum_j \int_{k_j} w_{eq} I \ \mathbf{ds} \end{aligned} \tag{5.9}$$

where the index J goes over the vertices that are at the junction of three kinematic elements. The index, b, goes over the vertices of the 1-D kinematic elements that are at the inflow boundary of the 2-D overland flow region. The index, e, goes over the vertices of the 1-D kinematic elements that are at the outflow boundary of the 2-D overland flow region. These are the vertices that are connected to the channel elements. We are using the same notation as in Section 4.3.1 for the upstream and the downstream elements at the junction. Notice that the sum of fluxes over all non-junction vertices is zero as usual, because of the definition of numerical flux. The sum of fluxes over all junction vertices, J, takes the form shown in (5.9) because at most three kinematic elements can meet at a junction.

For simplicity, we assume that there is no infiltration. At the inflow boundary of the overland flow region, we prescribe the condition $u = 0$. Then,

using the junction conditions (4.13) and (4.14), the sum of all fluxes over J reduces to 0. We can now reduce (5.9) to the following form:

$$\frac{d}{dt}\sum_j M_{of,j} = -\sum_e w_{eq}(Hu)_e + \sum_j \int_{k_j} w_{eq} R \ \mathbf{ds} \tag{5.10}$$

where $M_{of,j}$ is the water mass in overland flow element k_j. Notice that in the equation above, we have made the substitution $M_{of,j} \equiv \int_{k_j} w_{eq} H \ \mathbf{ds}$.

Now, we add (5.7) and (5.10) to obtain the following equation for the total water in our simulation domain:

$$\frac{d}{dt}\left(\sum_i M_{ch,i} + \sum_j M_{of,j}\right) = \sum_i \int_{s_i} q_L \ \mathbf{ds} - \sum_e w_{eq}(Hu)_e + \sum_j \int_{k_j} w_{eq} R \ \mathbf{ds} \tag{5.11}$$

Since q_L is constant over a channel element, we have:

$$\int_{s_i} q_L \ \mathbf{ds} = q_L L_i$$

Using the coupling condition (5.2) and the equation above, we get:

$$\int_{s_i} q_L \mathbf{ds} = w_{eq}(Hu)_{v_e} \tag{5.12}$$

Now, notice that $q_L = 0$ in the elements of the channel that do not receive flow from an overland flow element. Thus, the number of channel elements that receive flow from overland flow elements is the same as the number of vertices of the 1-D overland flow elements that are at the outflow boundary of the 2-D overland flow region. Using this insight and substituting (5.12) into (5.11), we obtain:

$$\frac{d}{dt}(TM) = \sum_j \int_{k_j} w_{eq} R \mathbf{ds} \tag{5.13}$$

where we have replaced the sum of the total mass of water in channels and the total mass of water in the overland flow region with TM, which represents the total mass of water in the entire domain. Thus, we see that the change of water mass in the domain is regulated only by the rainfall. The coupling scheme neither dissipates mass nor adds water mass, i.e. the coupling scheme is conservative.

5.2 Coupling of Channel Networks with a Bay

In a hurricane storm surge induced flooding simulation, the simulation domain often contains portions of a bay through which the simulation domain receives surge water. We simulate the flow dynamics in such region using DG-SWEM. DG-SWEM solves the two-dimensional shallow water equations in the domain using an adaptation of the Runge–Kutta local discontinuous Galerkin (RKLDG) methods developed by Cockburn and Shu [23] and modified by Cockburn and Dawson [22]. The details of the solution scheme can be found in a number of papers (See [5, 12, 26, 54–56]). The effects of a hurricane are incorporated into the simulation domain by incorporating external forces F_x and F_y in the x- and y- momentum equations respectively. These external force terms include forces like hurricane wind stress, that is usually computed using a standard quadratic drag law, and atmospheric pressure. These are the forces responsible for driving storm surge.

In our simulation, we couple the bay with the rest of the watershed through a channel. We discretize the bay with triangular elements such that

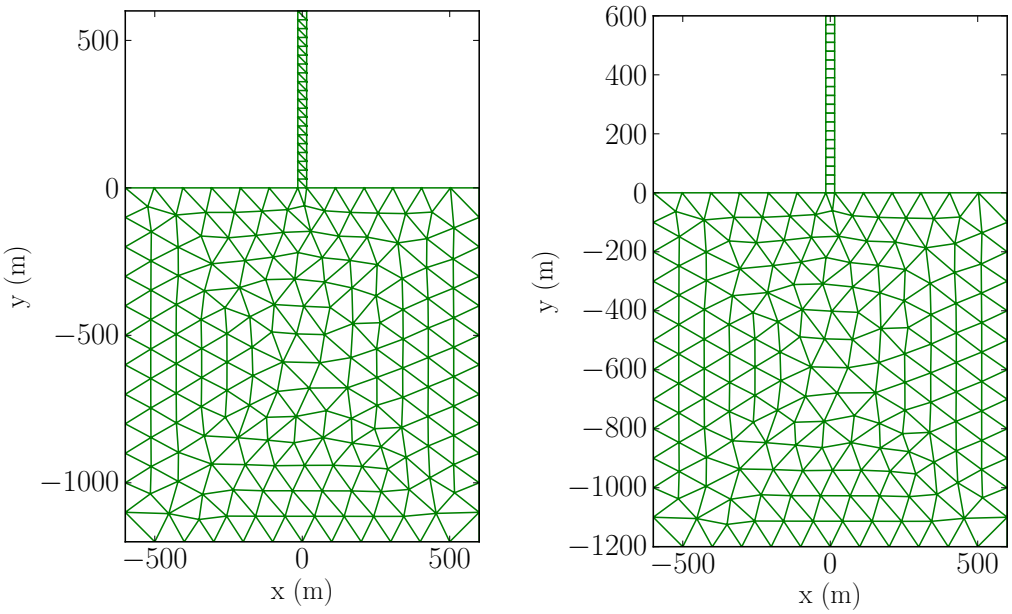
only one triangular element is connected to the channel, and the length of the edge connected to the channel is equal to the width of the channel at that point. The interface conditions developed for coupling 1-D channels with 2-D junctions (3.17a) – (3.17d) are imposed as the interface conditions to couple the channel with the bay as well. Even though we use DG-SWEM to simulate the flow dynamics in the bay, the channel and the bay are tightly coupled. We have modified a few DG-SWEM files to allow it to call a function that evolves the watershed to the next time step and to enforce the interface conditions on the bay side of the edge. We compile DG-SWEM (a FORTRAN code) and our watershed code (a C code) as a single executable and run it as a single program. The sharing of information between DG-SWEM and the watershed code happens at every time step through local memory, i.e. there are no i/o operations involved in passing the information between these two sub-domains. In the DG framework, the only information that needs to be exchanged between these two sub-domains are the values of the conserved variables at the interface edge of the bay and the interface vertex of the channel. Therefore, with a very little exchange of information, we can simulate the watershed domain and the bay as a single domain using a single executable. This coupling allows for a bidirectional exchange of information at each time step, and thus allows for a true coupling of rainfall and storm surge.

5.2.1 Numerical Simulations

We present a numerical simulation to verify that our coupling scheme does indeed allow for water to drain from the channel into the bay. We present another numerical simulation to verify that the surge water from the bay can indeed be propagated up the channel. For validation of the coupling scheme, we compare the results of both of these simulations to that obtained with a fully 2-D simulation using DG-SWEM in the entire domain. We use the same simulation domain for both of these simulations. The simulation domain consists of a 1440000 m^2 frictionless bay connected to a frictionless channel that is 600 m long and 30 m wide. The meshes for this domain are shown in Figure 5.5. The figure on the left shows how the entire domain is discretized as a 2-D domain. The figure on the right shows how the channel is connected to the bay. The rectangular boxes indicate that the discretization is one dimensional.

The channel and the bay are both flat in the first simulation. Initially we start with a dry domain. For the channel and bay coupled simulation, we prescribe a discharge of 27 m^3/s at the upper end of the channel. For the fully 2-D simulation, we prescribe a normal flow equal to 0.9 m^2/s on the upper end of the channel. Notice that this results in the same volumetric discharge for the coupled as well as the fully 2-D simulation. We prescribe a no-flow boundary condition on all edges of the bay that are not connected to the channel. We run the simulation for half an hour.

In Figure 5.6, we see that 30 seconds after the simulation begins, the



(a) Fully 2-D simulation

(b) Coupled channel and bay simulation

Figure 5.5: Meshes for the fully 2-D simulation and for the coupled channel and bay simulation shown for comparison.

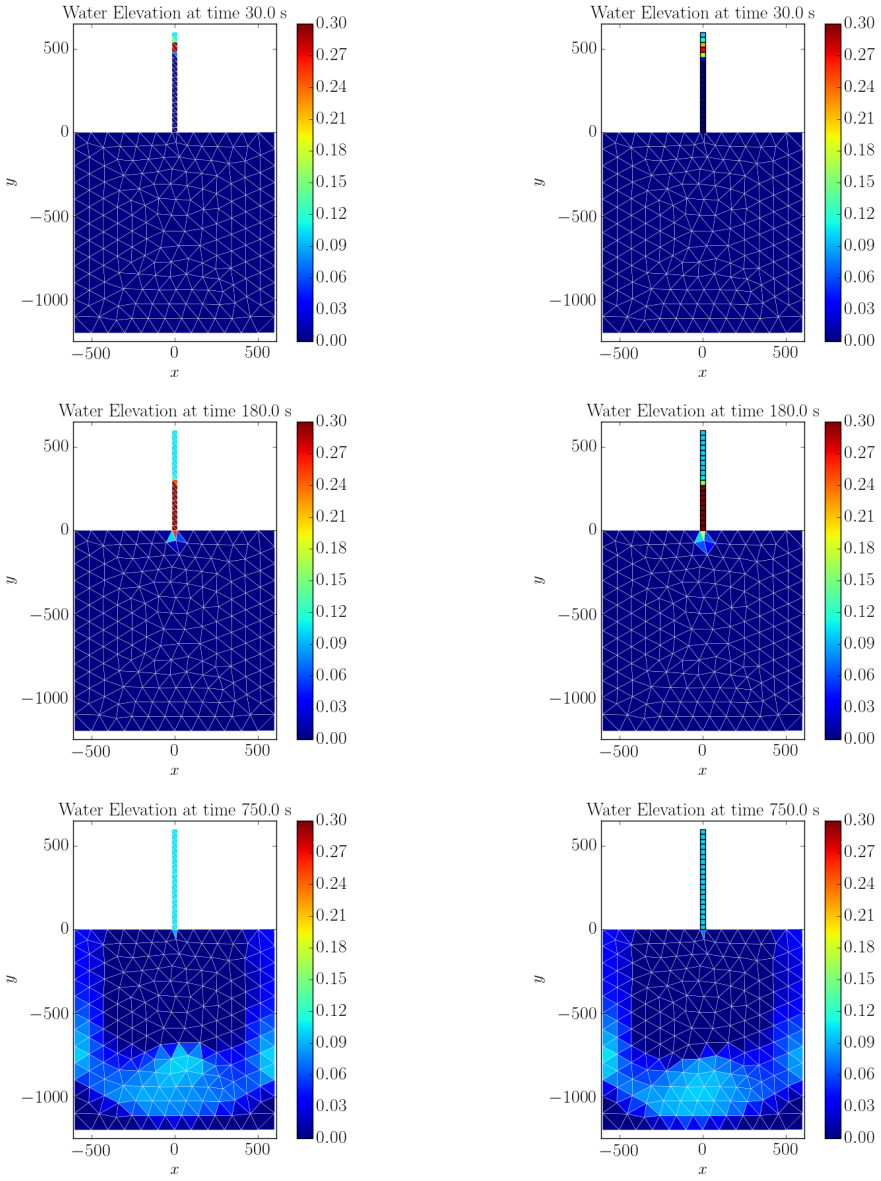


Figure 5.6: Water surface elevation obtained at different times using a fully 2-D simulation and a coupled channel and bay simulation. The domain is entirely flat and discharge is prescribed at the top end of the channel.

results of the coupled simulation and the fully 2-D simulation agree. Water from the channel enters the bay at the same time in both simulations at 180 seconds. When the water first enters the bay, the water level in the element that receives this water seems to be slightly higher in the 2-D simulation than in the coupled simulation. This might be because of the way wetting and drying is handled in the bay. While the edge that connects the channel to the bay is treated as an internal edge in the 2-D simulation, this edge is treated as a boundary edge in the coupled simulation. Thus, in the 2-D simulation, this edge is treated as a wetting interface, however it is not treated as a wetting interface in the coupled simulation. The water first hits the lower boundary of the bay and bounces back at the same time at 750 seconds in both simulations. The results of the two simulations are the same at this time. Thus, we have established that the coupling allows for an accurate transfer of information from the channel to the bay.

In the second simulation, the bay is flat, but the channel slopes upward away from the bay. The slope of the channel is 0.005. Initially we start with a dry domain. Water comes into the domain through the lower end of the bay at the rate of 0.2 m^2/s. We enforce a no-flow boundary condition on the upper end of the channel and all other sides of the bay. We run the simulation for half an hour. The results are shown in Figure 5.7. Observe that as the water comes into the bay from the lower end, in both simulations, it reaches the upper end of the bay for the first time at 240 seconds after the simulation begins. At this point we notice that some water has entered the

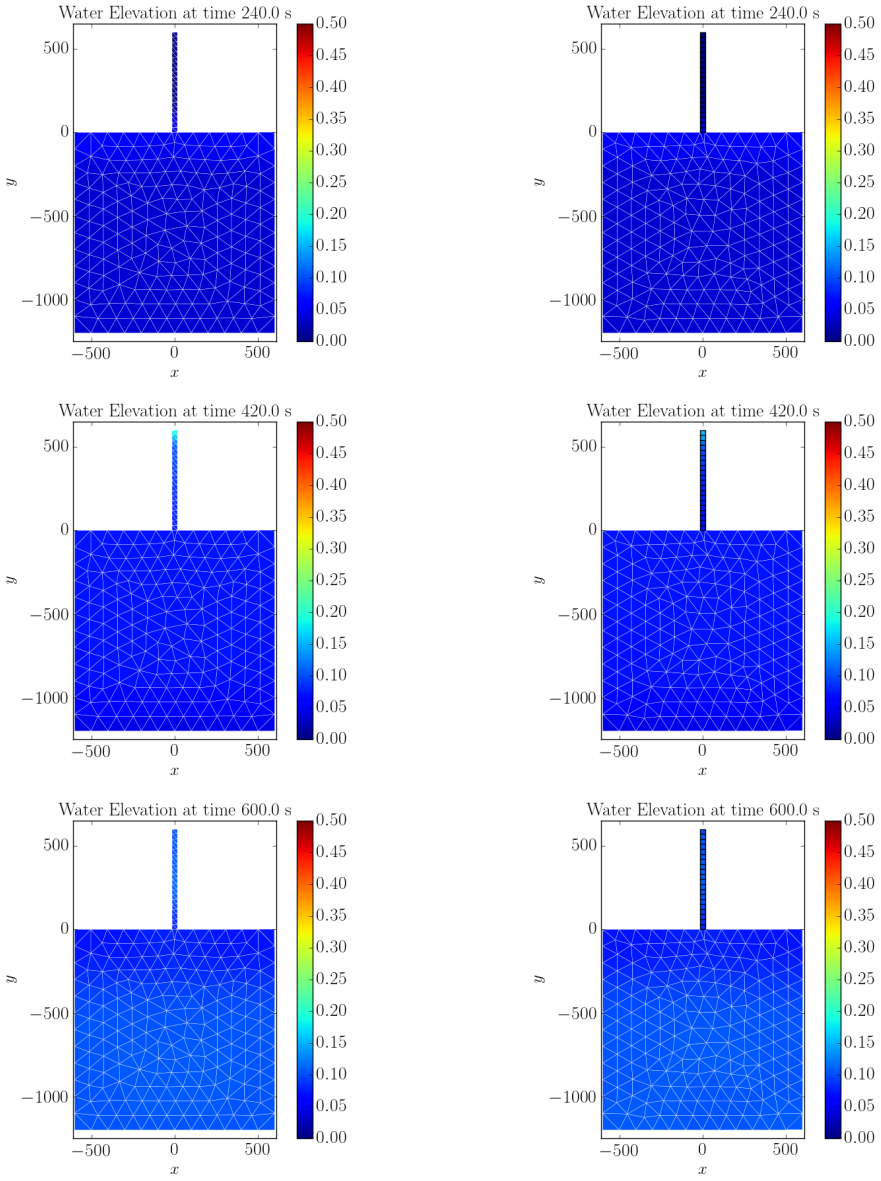


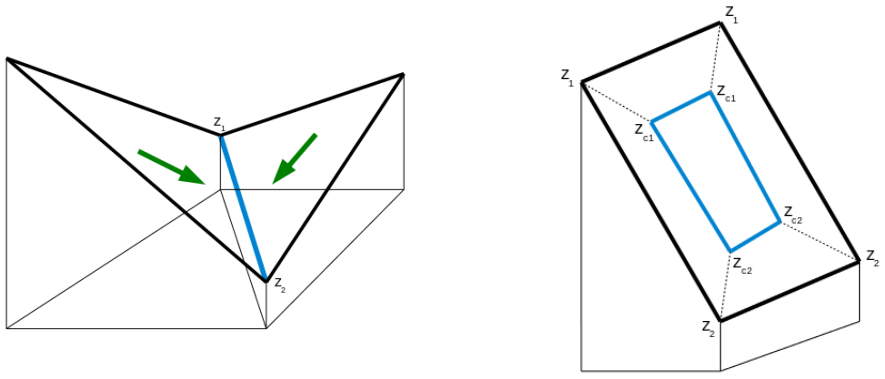
Figure 5.7: Water surface elevation obtained at different times using a fully 2-D simulation and a coupled channel and bay simulation. The channel slopes upward and discharge is prescribed at the bottom end of the bay.

channel and is traveling up the slope. This water hits the upper boundary of the channel in both simulations at 420 seconds, where we have prescribed a no-flow boundary condition. Because of the boundary condition, the water bounces back from the wall and travels down the channel again. We see that the water surface level in the channel has become uniform again at 600 seconds as the water traveling down the channel goes back into the bay again. Notice that the results of the fully 2-D simulation and the coupled simulations are very similar. The slight differences that appear in the values are artifacts of plotting values in rectangles and triangles using different software packages.

5.3 Flooding

Natural streams and rivers and artificial channels all have a certain carrying capacity associated with them. Once these channels and rivers get full either because of rainwater or surge water, the channels will flood their banks. This water will flow out of the channels laterally and will inundate the land areas adjacent to the channels. While, initially, we model these adjacent land areas as overland flow regions, once the channels have flooded, we can no longer model these regions as such. In order to handle the dynamics of flood waves and to allow the water to propagate against the direction of the surface slope, we convert these regions from overland flow to shallow water regions.

Now, we need to devise a way to determine when the channels are considered flooded. To this end, we first determine the carrying capacity of each channel element by using the elevation of the channel nodes and its banks.



(a) Representation of channel element in the mesh

(b) Illustration of a physical channel element and its bank

Figure 5.8: Channel Element

Recall that in a mesh of a watershed, a channel element is represented by the common edge shared by two overland flow elements as shown in Figure 5.8a. The elevations of the overland flow elements at the two vertices that form this edge are represented by z_1 and z_2 respectively. In Figure 5.8b, we illustrate what the channel element might look like physically. The blue rectangle in Figure 5.8b represents one channel element. The black lines represent edges of overland flow elements. Notice that this element now has a width. We assign the same elevation to the overland flow elements on both banks of the channel as demonstrated in the figure. The corresponding vertices on both banks of the channel have the same elevation z_1 and z_2 respectively. The elevation of the vertices of this channel element, represented by z_{c1} and z_{c2} respectively, are different from the elevations of the overland flow element. This elevation difference is what is used to calculate the carrying capacity of the channel at the beginning of the simulation. For simplicity, assume that the channel has

a rectangular cross-section with width W. Then, the carrying capacity of a channel element Ω_e, represented by V_{cap,Ω_e} is calculated as follows:

$$V_{cap,\Omega_e} = \left(\frac{|z_1 - z_{c1}|\, W + |z_2 - z_{c2}|\, W}{2} \right) |\Omega_e| \tag{5.14}$$

We calculate the total volume of water in a channel element at every time step as follows:

$$V_{\Omega_e} = \int_{\Omega_e} A \; \mathbf{ds} \tag{5.15}$$

The channel will be considered flooded if for any element Ω_e, $V_{\Omega_e} > V_{cap,\Omega_e}$. When this happens, we convert the overland flow region to shallow water region. And now, the shallow water region representing the floodplains and the channel has to be coupled bidirectionally. For this purpose, we let the excess water escape the channel laterally. Recall the channel continuity equation:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial \mathbf{s}} = q_L \tag{5.16}$$

Let the channel element Ω_e be connected to m floodplain elements. Notice that m is either 1 or 2. We now prescribe q_L in the following manner:

$$\begin{aligned} q_L &= q_{kin} && \text{if } V_{\Omega_e} \leq V_{cap,\Omega_e} \\ q_L &= q_{excess} + \sum_{i=1}^{m} Q^{in}_{n,fp,i} && \text{if } V_{\Omega_e} > V_{cap,\Omega_e} \\ &= \frac{V_{cap,\Omega_e} - V_{\Omega_e}}{|\Omega_e| \Delta t} + \sum_{i=1}^{n} Q^{in}_{n,fp,i} \end{aligned} \tag{5.17}$$

where q_{kin} represents the discharge that is received by the channel from the kinematic elements and $Q^{in}_{n,fp,i}$ represents the normal flow at the edge of the

i-th floodplain element connected to the channel element. Notice that q_{excess} is negative. We allow the water that has escaped the channel to come into the floodplains as a normal flow boundary condition on the floodplain edge as follows:

$$Q_{n,fp}^{boundary} = q_L \tag{5.18}$$

Once the channel starts flooding, we expect the water surface level in a channel element to equilibrate with the water surface level in the floodplain elements to which it is connected. Thus, we update the carrying capacity of the channel at every time step after flooding has occurred as follows:

$$V_{cap,\Omega_e} = V_{cap,\Omega_e}^{orig} + \frac{W}{m}\left(\sum_{i=1}^{m}\int_{\Omega_{fp,i}} H_{fp,i}\ \mathbf{ds}\right) \tag{5.19}$$

where V_{cap,Ω_e}^{orig} represents the carrying capacity of the channel element before it started flooding which is given in (5.14). $H_{fp,i}$ represents the water height in the i-th floodplain element.

Chapter 6

Simulation of Flooding in an Artificial Coastal Watershed

In this chapter, we present a comprehensive view of all the individual components of a coastal watershed flooding simulation that we have discussed so far. For this purpose, we have created an artificial watershed that contains all the sub-domains we have discussed so far. The mesh that is used to represent the watershed is shown in Figure 6.1. The z- coordinate represents the elevation of the watershed. The watershed contains a river network (channel network), whose vertices are represented by the red circles in the mesh. The watershed is connected to a bay through a channel segment at the vertex $(0, 0)$. The depth of the river network is not shown. The river network consists of three channel segments that comprise the main channel, three lateral channel segments and two junctions. Among the channel segments that comprise the main channel, the most upstream channel segment has a slope of 0.03, the one immediately downstream of this channel segment has a slope of 0.04 and the most downstream channel segment has a slope of 0.05. The lateral channel segments all have a slope of 0.02. The three channel segments that comprise the main channel are 4 m wide and the three lateral channel segments are 2 m wide. The river is shallowest at the upstream end and is deepest at the

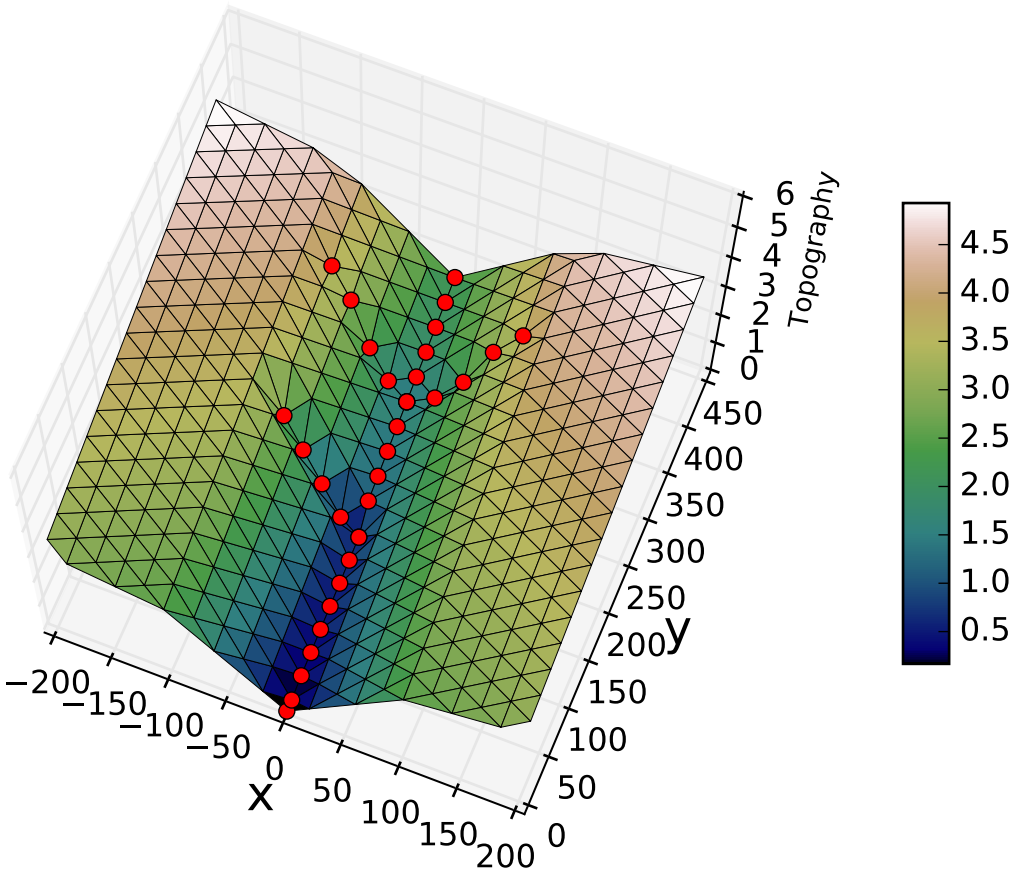


Figure 6.1: Artificial watershed containing a river network

downstream end where it is connected to the bay. Although we have assigned a uniform value for the Manning's friction coefficient, this value can be different at each vertex. The channel is also assumed to have the same friction coefficient as the hillslopes. We assume that the watershed is impermeable to water. Thus, we have turned infiltration off in our simulation. The bay is entirely flat at an elevation of 0 m and extends from -200 to 200 in the x-direction and -400 to 0 in the y- direction. It is assumed to be frictionless.

We run a simulation on the domain incorporating rainfall as well as storm surge. We assume that the bay is initially dry. The floodplains are dry as well. The channels and junctions are wet such that the water surface has an initial elevation of 0 m. The watershed, which does not include the bay, receives rainfall at an intensity of 3.15 in/hr for the first 20 minutes. At the same time, we allow surge water to come in from the bay in the form of a constant discharge imposed at the lowest boundary of the bay. Water comes into the bay at the rate of 2.5 m^2/s throughout the entire simulation. We enforce a no-flow boundary conditions on all edges of the channel that are not connected to either a junction or the bay. We run the simulation for half an hour.

In Figure 6.2, we have shown the flow directions that were calculated for the watershed using steepest descent (Approach II). The red vertices denote channel vertices. Notice that flow from elements on both sides of the edges connecting channel vertices to each other are directed to those edges. In the beginning of the simulation, the floodplains are treated as overland flow

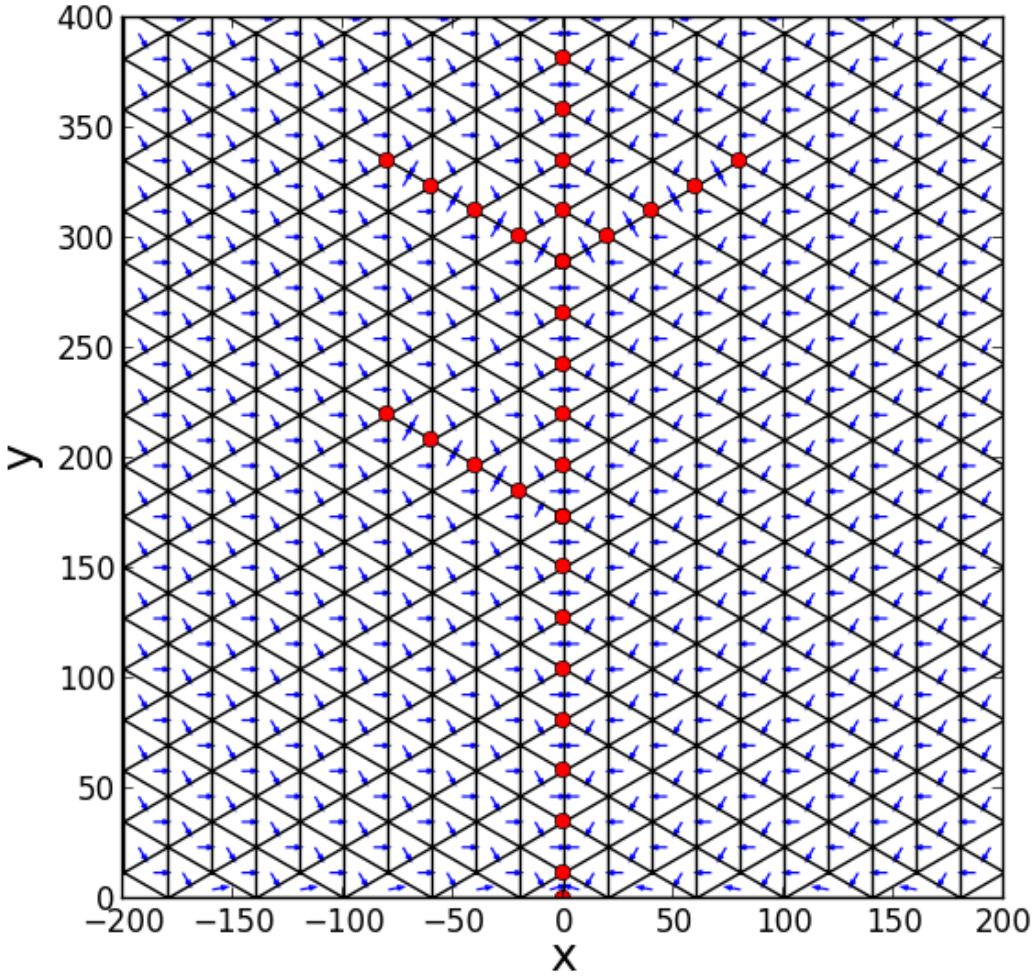


Figure 6.2: Flow directions calculated for the watershed along with the channel vertices

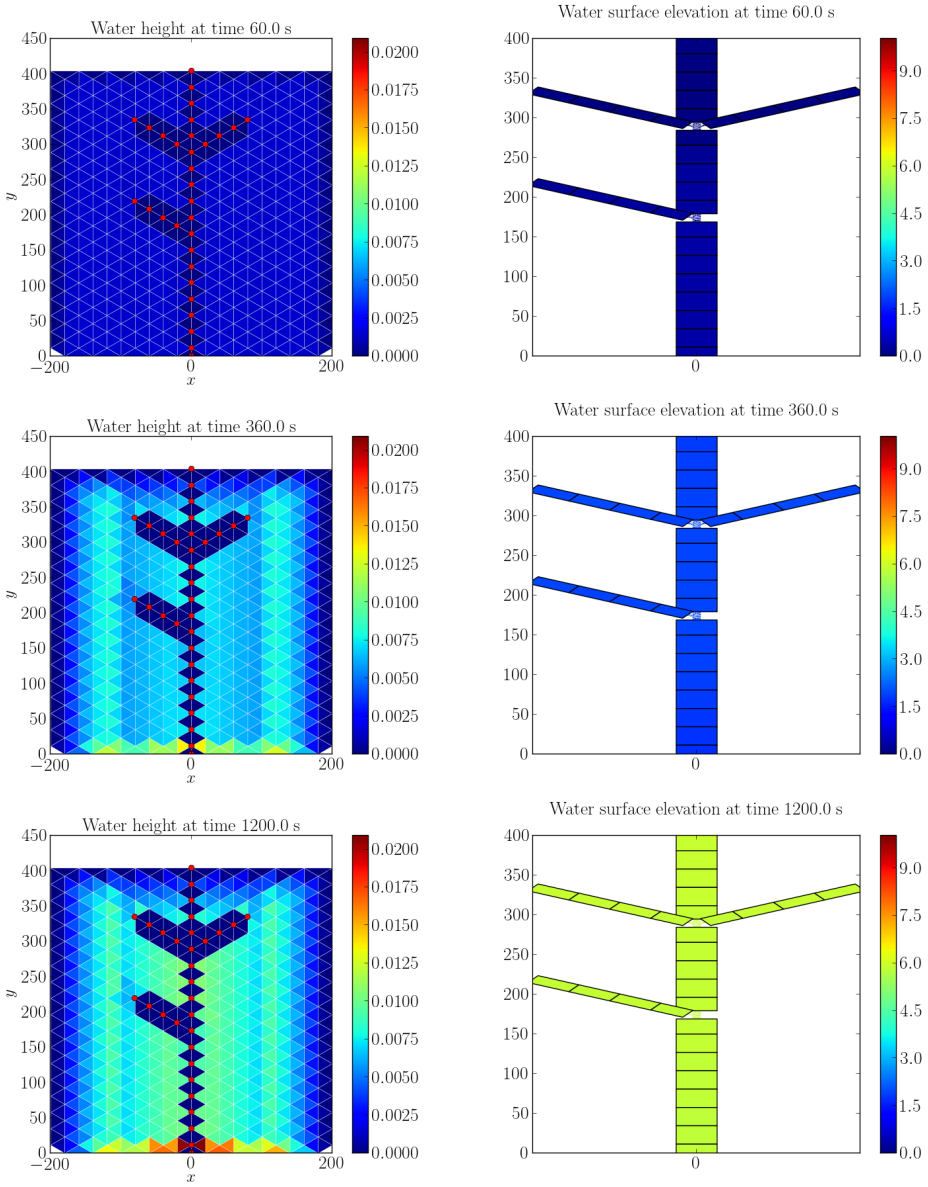


Figure 6.3: Water height in the floodplains (left) and water surface elevation in the channel network (right) obtained at different times during the simulation before flooding has occurred

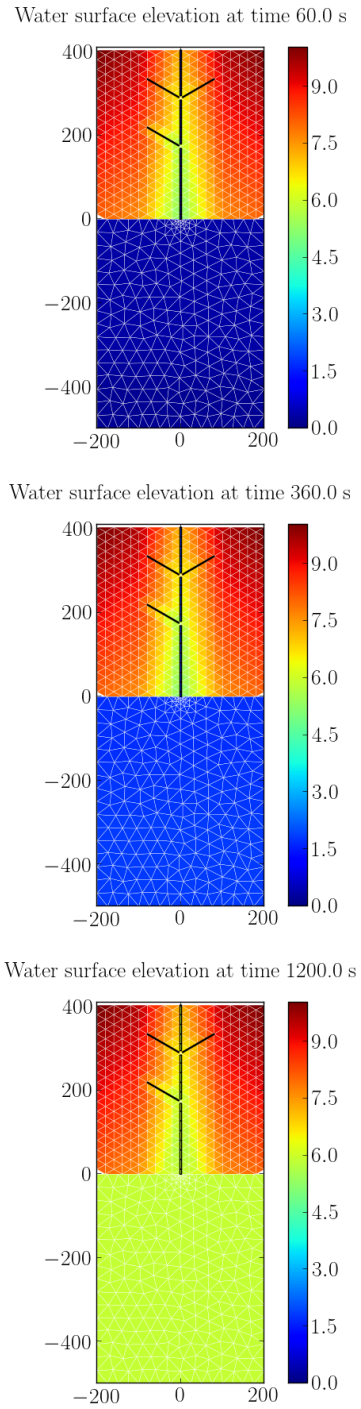


Figure 6.4: Water surface elevation in the bay before flooding has occurred

regions. The water height in the floodplains before the channels have started to flood are shown in Figure 6.3. We have plotted the water height instead of the water surface elevation in the floodplains because the water height is so low and the topography varies over such a large range that it will be hard to notice any water in the floodplains if the water surface elevation is plotted. The unchanging region of blue in the figures represent the channel network, which is not included in those figures, but is shown separately on the right. In the channel network, we have plotted the water surface elevation because the bottom of the channel network lies below the geoid. Therefore, any positive change in surface elevation is caused by the rise in the water level in the channel network. The channels have been enlarged in Figure 6.3 so that the change in the water levels in the channels are visible. The junctions, however, have been drawn to scale for reference.

Observe that, in the beginning of the simulation, the water in the floodplains is spread uniformly. This is because the floodplains receive rainfall uniformly over the domain. As the simulation continues, the water starts flowing down the floodplains towards the channels. Thus we see that the water levels in the floodplain are higher closer to the channel network. Notice that the water surface in the channel is rising as the simulation continues. In twenty minutes, the water level in the channel network has risen uniformly by about 6 m. This rapid rise of water in the channel network is happening not just because of the rainwater coming into the channel network from the floodplains, but also because of the surge water coming into the watershed from the bay.

We have shown the changes in water surface elevation in the bay because of surge water during this time period in Figure 6.4. Notice that even though the domain shown in Figure 6.4 does include the floodplains, the rainwater flowing down these floodplains are not shown in these pictures.

The rain stops after 20 minutes. At this point, the water in some of the channels has almost exceeded the carrying capacity of the channels. Thus, the channels start flooding. At this point, we change the floodplains from an overland flow region sub-domain to a shallow water region sub-domain. The water surface level in the entire domain at various stages of flooding is shown in the left column in Figure 6.5. The right column shows the water surface level in the channel network. The channels are again enlarged, while the junctions are drawn to scale. Notice that as the channels start flooding, water flows out of the channels laterally. The water surface level in the channel network is level with the water surface level in the bay. We see that at time 1500 s or 25 minutes after the simulation has begun, water has started moving up the floodplain against the direction of the surface slope. By the time we stop the simulation at 1800 s or half an hour after the simulation has started, the entire watershed is at the same elevation. Parts of the watershed near the exterior boundary that had the highest topography are still dry. However, the entire watershed will be inundated if we run the simulation longer under the same conditions.

Thus, with this example, we have demonstrated how all of the sub-domains and the coupling techniques that we have discussed in this dissertation

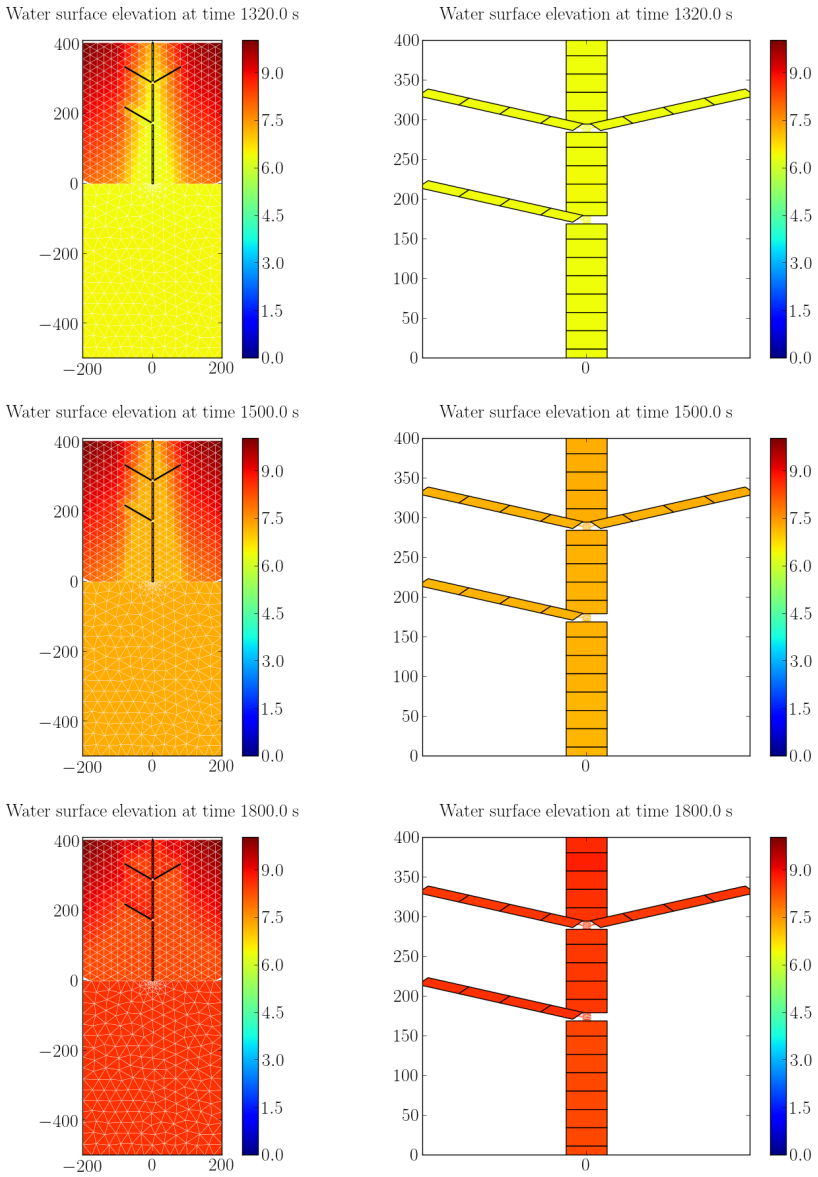


Figure 6.5: Water surface elevation in the floodplains and the bay (left) and water surface elevation in the channel network (right) obtained at different times during the simulation after flooding has occurred

come together to form a comprehensive way of simulating flooding in a coastal watershed.

Chapter 7

Conclusion

We undertook this research with the aim of developing an inclusive framework for modeling flooding in coastal watersheds. Most of the current research efforts in the arena have focused either on the watershed part, using storm surge as a boundary condition that does not interact with rainfall, or they have focused on the coastal side using river inflow as a boundary condition. Our goal was to improve on these efforts by devising a technique to truly allow rainfall to interact with storm surge. Herein lies the novelty of our work.

We have developed a way to decompose a watershed containing hillslopes/floodplains, rivers and parts of the ocean into various sub-domains depending on the dynamics of the flow. The various sub-domains are: overland flow region, channel network and shallow water region. We model flow in the entire domain in the context of shallow water theory. However, wherever appropriate, like in the overland flow regions, we use a simplified form of the shallow water equations called the kinematic wave equation. When the dynamics of flow in the region becomes such that it cannot be handled by the kinematic wave equation, we convert the sub-domain to shallow water region and model flow in the region with the full 2-D shallow water equations. In

order to resolve sub-grid scale features like rivers and streams in the domain, we represent river segments as 1-D domains that are coupled to each other via 2-D junctions. This modeling framework allows us to gain computational efficiency without compromising on physical accuracy.

We solve the 1-D shallow water equations, the 2-D shallow water equations and the kinematic wave equation using a discontinuous Galerkin method. Using this framework, we have shown that the coupling between various domains can be done very naturally through the numerical flux. Specifically, during the calculation of the numerical flux on an interface edge of a sub-domain, we prescribe the value of the conserved variables arising from the connected sub-domain on that interface edge as the exterior state of that variable. This exchange of information is done at every time step. Thus each sub-domain receives dynamic information in time from the neighboring sub-domain.

We established the validity of our representation of a channel network and the way we couple channels and junctions through comparisons with experimental results, comparisons with other junction models and comparisons with a fully 2-D simulation. We analytically verified that the 1-D/2-D coupling that happens in the channel network is mass conservative. Using a simplified form of the shallow water equations and an approach similar to the one taken in [4], we also proved that the 1-D/2-D coupling is stable. We have developed a technique similar to the one described in [92] to simulate flow in a two dimensional region using the 1-D kinematic wave equations. Analytical solutions of the kinematic wave equation with rainfall of constant intensity and zero infil-

tration are available. We treated a parking lot of unit width as a 2-D domain and simulated flow induced by rainfall of constant intensity on the region. Through comparisons with the analytical solution, we established the validity of our approach. We verified that there is no water mass being lost in the coupling of overland flow region with the channel network analytically as well as numerically. We have established the validity of the coupling between the channel network and the shallow water region (bay) through comparisons of the simulation done using our modeling framework with a fully 2-D simulation done on the entire domain using DG-SWEM. We developed a way to simulate the flow dynamics in the watershed once the channels flood. This involved developing a way to couple the channels laterally with floodplains. Finally, we simulated an entire watershed containing all the individual components we described to provide a proof of concept of the whole modeling framework.

A big attraction of our modeling framework is the potential for computational gain because of simulating certain parts of the domain as 1-D domains and solving a simpler equation on other parts. For example, in overland flow regions and in parts of the bay further away from the coast where large discretizing elements are used, much larger time steps can be taken than elsewhere in the domain. In our work so far, we have not exploited this fact. Future work should include implementing sub-domain time stepping techniques so that we can allow each sub-domain to take the largest possible time step without losing stability. In the future, we will also employ the modeling framework we have developed here to hindcast some real flooding events induced by hurri-

cane storm surge. This will involve adapting the watershed code to use DEM (digital elevation model) data of a real watershed to calculate flow paths. It will also require addition of some domain preprocessing routines to fill sinks in the domain. A more sophisticated routine to calculate the flow paths and determine the stream networks needs to be developed. Similar to other such softwares, we will have to develop a way to burn in the stream network data available for real watersheds. With these additions, this software package we have developed has a potential to bridge the gap between the watershed hydrology models and coastal hydrodynamic models.

Appendices

Appendix A

Derivation of the 1-D Saint Venant Equations

Although the 1-D Saint Venant Equations can be derived from the Navier–Stokes equation, we provide a more intuitive derivation here based on the basic principles of conservation of mass and momentum. The derivation here closely follows the derivation given in [25]. The fundamental assumptions underlying the development of the Saint Venant equations are the following:

1. The wave surface varies gradually along the length of the channel. This assumption allows us to assume that vertical accelerations can be neglected and that the pressure distribution along a vertical is hydrostatic.

2. Friction losses in non-uniform, unsteady flow are not significantly different from those in uniform, steady flow. This assumption allows us to use the Manning and Chezy equations to describe channel resistance. These equations were initially developed for steady, uniform flow conditions.

3. The flow is one-dimensional. The water depth and flow velocity vary only in the direction of flow. Therefore, the flow velocity is constant and the water surface is horizontal across any section perpendicular to the direction of flow.

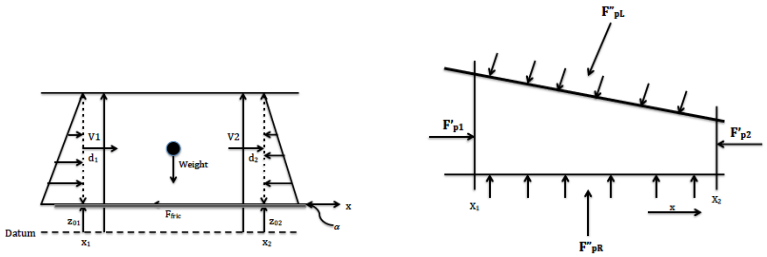
4. The channel bed slope is small, so that $sin(\alpha)$ may be replaced by $tan(\alpha)$, which may be replaced by the bed slope and $cos(\alpha)$ may be replaced by unity. Here, α is the angle made by the channel bottom with the horizontal.

5. The channel bed is stable such that there is no change in bed elevations in time.

6. The fluid is incompressible and of constant density throughout the flow.

With the assumptions listed above, we derive the 1-D Saint Venant equations, in which the dependent variables are the wet cross-sectional area and the discharge. The following symbols will be used in the derivation:

- x = co-ordinate in the longitudinal direction
- y = co-ordinate in the vertical direction
- t = time
- A = the area of the wet cross-section at the section (L^2)
- Q = the volumetric discharge at the section (L^3/T)
- V = mean velocity in the direction of flow at the section (L/T)
- d = total depth (or height) of water (L)
- $W = W(x, y)$ width of the channel at a depth y in the section (L)



(a) Side view (b) top view

Figure A.1: Control volume of a channel ($\Delta x = (x_1, x_2)$)

- g = acceleration due to gravity (L/T^2)
- ρ = mass density of the fluid (M/L^3)
- q_L = lateral inflow rate per unit length of the channel (L^2/T)

Note that the co-ordinate system is not entirely orthogonal, i.e. x lies in the bed of the channel and z is the vertical co-ordinate. However, this arrangement works because of assumption 4, which says that the cosine of the bed slope is approximately unity.

Now, consider a control volume (Δx) of a channel extending from $x = x_1$ to $x = x_2$ as shown in Figure A.1 for a time period $\Delta t = t_2 - t_1$.

A.1 Derivation of the continuity equation

The balance law for total mass in a control volume states that the net mass flux into a control volume equals the net mass increase of the control

volume. Now, the net mass flux into Δx over a time period Δt is given by:

$$\int_{t_1}^{t_2} (\rho\, V_1 A_1 - \rho\, V_2 A_2)\, \mathrm{dt} + \int_{t_1}^{t_2} \int_{x_1}^{x_2} \rho\, q_L \mathrm{dx}\ \mathrm{dt}$$

The net increase in mass in Δx from time $t = t_1$ to time $t = t_2$:

$$\int_{x_1}^{x_2} ((\rho\, A)_{t_2} - (\rho\, A)_{t_1}) \mathrm{dx}$$

Thus the balance law for mass states:

$$\int_{t_1}^{t_2} (\rho\, V_1 A_1 - \rho\, V_2 A_2)\, \mathrm{dt} + \int_{t_1}^{t_2} \int_{x_1}^{x_2} \rho\, q_L \mathrm{dx}\ \mathrm{dt} = \int_{x_1}^{x_2} ((\rho\, A)_{t_2} - (\rho\, A)_{t_1})\, \mathrm{dx}$$

$$\int_{t_1}^{t_2} (\rho\, V_1 A_1 - \rho\, V_2 A_2)\, \mathrm{dt} + \int_{x_1}^{x_2} ((\rho\, A)_{t_1} - (\rho\, A)_{t_2})\, \mathrm{dx} = -\int_{t_1}^{t_2} \int_{x_1}^{x_2} \rho\, q_L \mathrm{dx}\ \mathrm{dt}$$

Since ρ is constant, we can divide the above equation through by $-\rho$ to obtain:

$$\int_{t_1}^{t_2} (V_2 A_2 - V_1 A_1)\, \mathrm{dt} + \int_{x_1}^{x_2} (A_{t_2} - A_{t_1})\, \mathrm{dx} = \int_{t_1}^{t_2} \int_{x_1}^{x_2} q_L \mathrm{dx}\ \mathrm{dt}$$

Substituting $Q \equiv VA$ in the above equation, we obtain:

$$\int_{t_1}^{t_2} (Q_2 - Q_1)\, \mathrm{dt} + \int_{x_1}^{x_2} (A_{t_2} - A_{t_1})\, \mathrm{dx} = \int_{t_1}^{t_2} \int_{x_1}^{x_2} q_L \mathrm{dx}\ \mathrm{dt} \tag{A.1}$$

Now, the fundamental theorem of calculus states that

$$A_{t_2} - A_{t_1} = \int_{t_1}^{t_2} \frac{\partial A}{\partial t}\, \mathrm{dt} \tag{A.2}$$

Similarly:

$$Q_2 - Q_1 = \int_{x_1}^{x_2} = \frac{\partial Q}{\partial x}\, \mathrm{dx} \tag{A.3}$$

Now, substituting (A.2) and (A.3) into (A.1), we obtain the following:

$$\int_{x_1}^{x_2} \int_{t_1}^{t_2} \left(\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} \right)\ \mathrm{dt}\ \mathrm{dx} = \int_{t_1}^{t_2} \int_{x_1}^{x_2} q_L \mathrm{dx}\ \mathrm{dt} \tag{A.4}$$

Since (A.4) is valid over an infinitely small area (dxdt), we obtain the following continuity equation:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = q_L \tag{A.5}$$

A.2 Derivation of the momentum equation

The balance law for momentum in a control volume states that the net change of momentum in the x-direction in the control volume equals the sum of the momentum flux into the control volume and the forces applied to the control volume in the x-direction. Mathematically, the momentum balance law states:

$$\int_{x_1}^{x_2} ((\rho V A)_{t_2} - (\rho V A)_{t_1})\ \mathrm{dx} = \int_{t_1}^{t_2} ((\rho V^2 A)_1 - (\rho V^2 A)_2)\ \mathrm{dt} + \sum_i F_i \quad \text{(A.6)}$$

where the first term on the left hand side denotes the net flux of momentum into Δx over the time period Δt and the second term on the left hand side denotes the net change of momentum in Δx from time $t = t_1$ to time $t = t_2$. Thus the left hand side as a whole denotes the net change of momentum. The right hand side denotes the sum of the body forces applied to the control volume.

There are five different forces acting on the control volume:

1. The pressure forces at x_1 and x_2:

   These pressure forces represent hydrostatic pressure. They are denoted by F'_{p1} and F'_{p2} in Figure A.1b. The net pressure force resulting from these pressures (F_1) can be calculated as:

   $$F_1 = \int_{t_1}^{t_2} (F'_{p1} - F'_{p2})\ \mathrm{dt} = \int_{t_1}^{t_2} ((\rho g I_1)_1 - (\rho g I_1)_2)\ \mathrm{dt}$$

   where: $I_1 = \int_0^d (d - y) W\ dy$

2. The pressure force components on the channel sidewalls if the channel width varies in the x-direction (F_2):

   If the channel narrows or widens in the downstream direction, the walls contribute an additional pressure force in addition to the shear force which will be accounted for in F_4. These forces are denoted F''_{pL} and F''_{pR} in Figure A.1b. The net pressure force resulting from these pressures can be calculated as:

$$F_2 = \int_{t_1}^{t_2} (F''_{pL} - F''_{pR})\mathrm{dt} = \int_{t_1}^{t_2} \int_{x_1}^{x_2} \rho g I_2 \text{ dx dt}$$

   where $I_2 = \int_0^d (d-y)\left(\frac{\partial W}{\partial x}\right) \mathrm{dy}$

3. The gravity force component in the x-direction (F_3):

   The gravity force acting on the control volume is a function of the volume of the fluid, $dV = Adx$. The corresponding weight of the fluid can be expressed as: $W = \rho\, dV\, g = \rho g A dx$, where g is the gravitational acceleration. The component of the weight in the direction of flow becomes the gravity force: $F_g = \rho\, g\, A\, dx\, sin\theta$, where θ is the angle of inclination of the channel bed. With the assumption of a small angle of inclination of the channel bed, the sine of the angle can be approximated as the tangent of the angle: $sin(\theta) \approx tan(\theta)$ which is equal to the slope of the channel bed, S_0. Therefore, for a small angle of inclination of the channel bed, gravity force acting on the control volume can be written as:

$$F_3 = \int_{t_1}^{t_2} \int_{x_1}^{x_2} \rho g A S_0 \text{ dx dt}$$

4. The flow resistance due to friction (F_4):

   The frictional resistance is manifested by means of a shear along the bottom and sides of the channel. On the control volume Δx, this force is given by:

$$F_4 = \int_{t_1}^{t_2} F_{fric} \text{ dt} = -\int_{t_1}^{t_2} \int_{x_1}^{x_2} \rho g A S_f \text{ dx dt}$$

   where S_f is the friction slope that can be approximated by empirical relations like Manning or Chezy equations.

5. the contraction/expansion force due to abrupt changes in the channel cross section (F_e)

6. the wind shear force (F_w)

Now, substituting all of the expressions for the forces described above into the momentum equation (A.6), and dividing through by ρ, we obtain:

$$\begin{aligned}\int_{x_1}^{x_2} ((VA)_{t_2} - (VA)_{t_1}) \text{ dx} = &\int_{t_1}^{t_2} ((V^2A)_1 - (V^2A)_2) \text{ dt} + \int_{t_1}^{t_2} ((gI_1)_1 - (gI_1)_2) \text{ dt} \\ &+ \int_{t_1}^{t_2} \int_{x_1}^{x_2} gI_2 \text{ dx dt} + \int_{t_1}^{t_2} \int_{x_1}^{x_2} gAS_0 \text{ dx dt} \\ &- \int_{t_1}^{t_2} \int_{x_1}^{x_2} gAS_f \text{ dx dt}\end{aligned} \tag{A.7}$$

Let $Q \equiv VA$ and substituting this into the equation above, we obtain:

$$\begin{aligned}\int_{x_1}^{x_2}((Q)_{t_2}-(Q)_{t_1})\ \mathrm{dx} = &\int_{t_1}^{t_2}\left(\left(\frac{Q^2}{A}\right)_1-\left(\frac{Q^2}{A}\right)_2\right)\ \mathrm{dt}+\int_{t_1}^{t_2}((gI_1)_1-(gI_1)_2)\ \mathrm{dt}\\ &+\int_{t_1}^{t_2}\int_{x_1}^{x_2}gI_2\ \mathrm{dx}\ \mathrm{dt}+\int_{t_1}^{t_2}\int_{x_1}^{x_2}gAS_0\ \mathrm{dx}\ \mathrm{dt}\\ &-\int_{t_1}^{t_2}\int_{x_1}^{x_2}gAS_f\ \mathrm{dx}\ \mathrm{dt}\end{aligned} \tag{A.8}$$

Now, using the fundamental theorem of calculus as was done in (A.2) and (A.3), we get:

$$\begin{aligned}Q_2-Q_1 &= \int_{x_1}^{x_2}\frac{\partial Q}{\partial x}\ \mathrm{dx}\\ Q_{t_2}-Q_{t_1} &= \int_{t_1}^{t_2}\frac{\partial Q}{\partial t}\ \mathrm{dt}\\ \left(\frac{Q^2}{A}\right)_2-\left(\frac{Q^2}{A}\right)_1 &= \int_{x_1}^{x_2}\frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right)\ \mathrm{dx}\\ (I_1)_2-(I_1)_1 &= \int_{x_1}^{x_2}\frac{\partial gI_1}{\partial x}\ \mathrm{dx}\end{aligned} \tag{A.9}$$

Substituting the expressions in (A.9) into (A.8), we obtain:

$$\int_{x_1}^{x_2}\int_{t_1}^{t_2}\frac{\partial Q}{\partial t}\ \mathrm{dt}\ \mathrm{dx} = \int_{x_1}^{x_2}\int_{t_1}^{t_2}\left(-\frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right)-g\left(\frac{\partial I_1}{\partial x}-I_2-(S_0-S_f)A\right)\right)\ \mathrm{dt}\ \mathrm{dx} \tag{A.10}$$

Since (A.10) is valid over an infinitely small area (dx dt), we obtain:

$$\frac{\partial Q}{\partial t}+\frac{\partial}{\partial x}\left(\frac{Q^2}{A}+gI_1\right) = gI_2+gA(S_0-S_f) \tag{A.11}$$

(A.5) and (A.11) together are called the 1-D Saint Venant equations.

Appendix B

Numerical tests for the solution of the 1-D and the 2-D shallow water equations

Here, we test the performance of our codes for the 1-D and the 2-D shallow water equations on numerous test cases for the shallow water equations that are compiled in [27]. In the first five cases, we test the ability of the code to reproduce steady state solutions correctly by running the simulation for a large t. Then, we test the ability of the code to handle wetting and drying situations. In all of the numerical examples discussed, unless otherwise noted, the size of the time step is dynamically determined using CFL number equal to $\frac{1}{3}$, and the slope limiter is turned on. Additionally, unless otherwise noted, the water height is reported in meters and the discharge is given in m^3/s.

B.1 Flow over a bump

In this section, we test the ability of our code to handle the effect of bathymetry in the interior of the domain. By varying the boundary conditions, we also test how robust the code is in handling different flow regimes, depending on which the solution can either be smooth or constitute a shock. The domain for the three tests that follow is identical and can be described

by:

Domain:

$$0 \text{ m} < x < 25 \text{ m}$$

$$0 \text{ m} < y < 1 \text{ m}$$

Bathymetry:

$$z(x,y) = \begin{cases} -0.2 + 0.05(x-10)^2 & 8 \text{ m} < x < 12 \text{ m} \\ 0 & \text{else} \end{cases}$$

Manning's n:

$$0 \text{ (no friction)}$$

For the 1-D code, we model this region as a 1-D channel with a rectangular cross-section and a uniform width of 1m. We discretize the domain with uniform $h = 0.125$ m. For the 2-D code, we have a slightly coarser discretization in the y-direction than in the x-direction. We discretize the mesh with $h_{y,max} = 0.33$ m and with $h_{x,max} = 0.125$ m. For each of the cases below, we show the solution obtained from the 1-D code alongside the solution obtained from the 2-D code. Since the solution is uniform in the vertical direction, we only plot the solution along the horizontal cross-section of the channel. The analytical solutions plotted are the solutions to the 1-D St. Venant equations and are obtained from [27].

B.1.1 Smooth Subcritical Flow

The initial and boundary conditions we used to obtain a subcritical flow regime are:

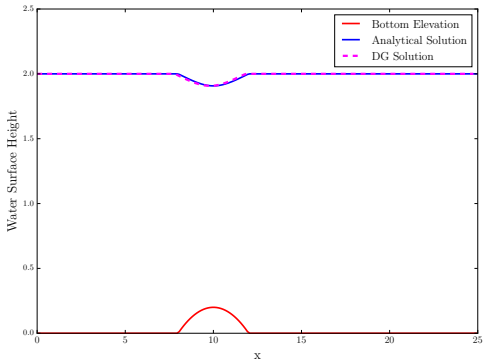
Initial Conditions:

Surface elevation:	$\zeta = 2$ m	
Flux:	$Q^{1D}n = 0$ m$^3/s$	$\mathbf{Q}^{2D} \cdot n = 0$ m$^2/s$

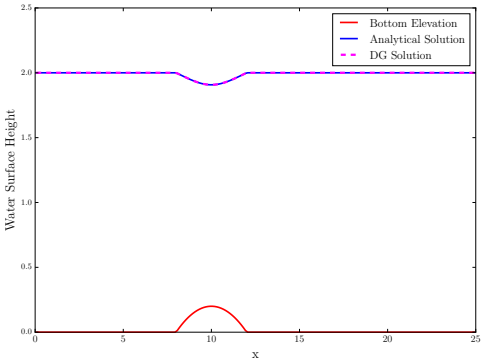
Boundary Conditions:

@x = 0:	$Q^{1D}n = 4.42$ m$^3/s$	$\mathbf{Q}^{2D} \cdot \mathbf{n} = 4.42$ m$^2/s$
@x = 25:	$\zeta = 2$ m	

Note that since the width of the channel is 1 m, $\mathbf{Q}^{2D} \cdot \mathbf{n} = Q^{1D}n$. As we can see in Figure B.1, the 1-D and the 2-D codes both perform exceedingly well in reproducing a steady state subcritical flow. The solutions calculated with both codes lie directly on top of the analytical solution.



(a) 1D solution



(b) 2D Solution along a horizontal cross-section

Figure B.1: Steady state smooth subcritical flow

B.1.2 Smooth Transcritical Flow

The initial and boundary conditions we used to obtain a transcritical flow regime with a smooth solution are:

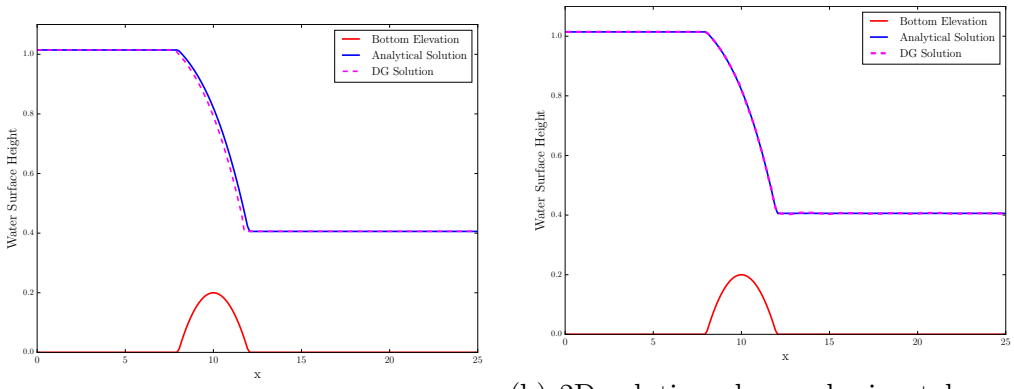
Initial Conditions:

Surface elevation:	$\zeta = 0.66$ m	
Flux:	$Q^{1D}n = 0$ m$^3/s$	$\mathbf{Q}^{2D} \cdot n = 0$ m$^2/s$

Boundary Conditions:

@x = 0:	$Q^{1D}n = 1.53$ m$^3/s$	$\mathbf{Q}^{2D} \cdot \mathbf{n} = 1.53$ m$^2/s$
@x = 25:	$\zeta = 2$ m	

As we can see in Figure B.2, both the 1-D and the 2-D codes reproduce



(a) 1D solution

(b) 2D solution along a horizontal cross-section

Figure B.2: Smooth transcritical flow

the steady state transcritical flow solution well. The 2-D solution seems to be better than the 1-D solution, however, the differences in the values of the

analytical solution and the 1-D numerical solution are on the order of 10^{-3} and are thus negligible.

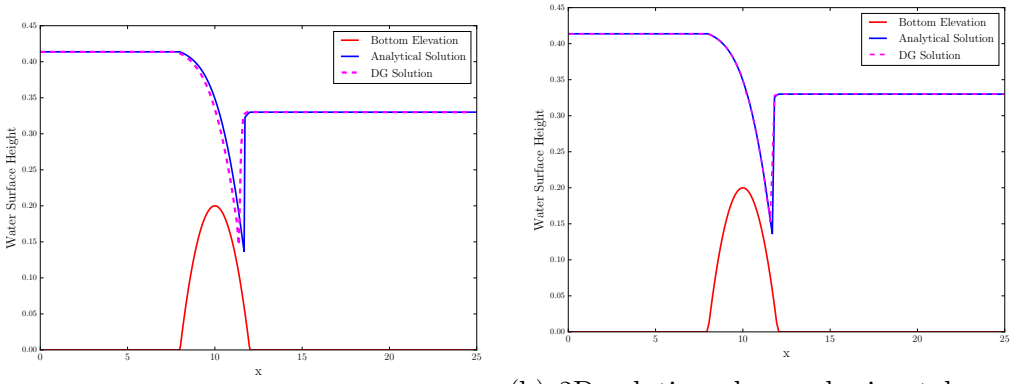
B.1.3 Transcritical flow with a shock

The initial and boundary conditions we used to obtain a transcritical flow regime with a shock are: **Initial Conditions:**

Surface elevation:	$\zeta = 0.33$ m	
Flux:	$Q^{1D}n = 0$ m$^3/s$	$\mathbf{Q}^{2D} \cdot n = 0$ m$^2/s$

Boundary Conditions:

@x = 0:	$Q^{1D}n = 0.18$ m$^3/s$	$\mathbf{Q}^{2D} \cdot \mathbf{n} = 0.18$ m$^2/s$
@x = 25:	$\zeta = 0.33$ m	



(a) 1D solution

(b) 2D solution along a horizontal cross-section

Figure B.3: Transcritical flow with a shock

We can see in Figure B.3 that the 1-D solution computes higher values

than the analytical solution at the shock. The 2-D solution captures shocks more accurately.

B.2 MacDonald's short channel with smooth transition and shock

In this case, we test the treatment of the friction term. Since the bathymetry varies near the boundary of the domain, we get a better validation of the boundary term as well. The flow is subcritical at inflow, smoothly transitions to being supercritical, and becomes subcritical again through a shock. The domain is described by:

Domain:

$$0 \text{ m} < x < 100 \text{ m}$$

$$0 \text{ m} < y < 1 \text{ m}$$

Manning's n:

$$0.0328 \text{ m}^{-\frac{1}{3}} s$$

For the 1-D code, we model this region as a 1-D channel with a rectangular cross-section and a uniform width of 1 m. We discretize the domain with uniform $h = 0.5$ m. For the 2-D code, we have a coarser discretization in the y-direction than in the x-direction. We discretize the mesh with $h_{y,max} = 0.33$ m and $h_{x,max} = 0.5$ m. For each of the cases below, we show the solution obtained from the 1-D code alongside the solution obtained from the 2-D code. Since the solution is uniform in the vertical direction, we only

plot the solution along the horizontal cross-section of the channel. The analytical solution plotted is the solution to the 1-D St. Venant equations. The bathymetry and the analytical solution are obtained from [27].

The initial and the boundary conditions used for this simulation are: **Initial Conditions:**

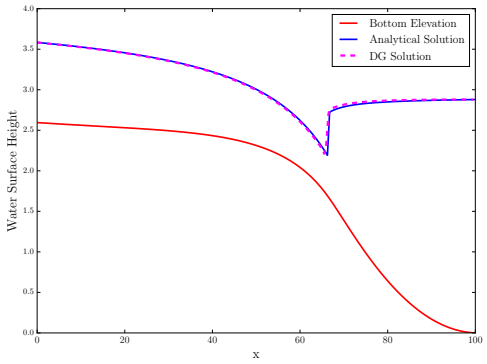
Water Height: $H(x, y) = \max(2.87928 - z(x), 0)$ m

Flux: $Q^{1D}n = 0$ m$^3/s$ $\mathbf{Q}^{2D} \cdot n = 0$ m$^2/s$

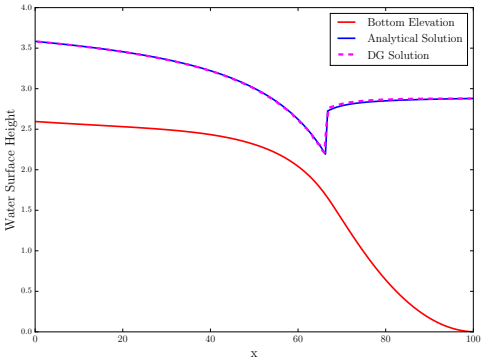
Boundary Conditions:

@x = 0: $Q^{1D}n = 2$ m$^3/s$ $\mathbf{Q}^{2D} \cdot \mathbf{n} = 2$ m$^2/s$

@x = 100: $H = 2.87844$ m



(a) 1D solution



(b) 2D solution along a horizontal cross-section

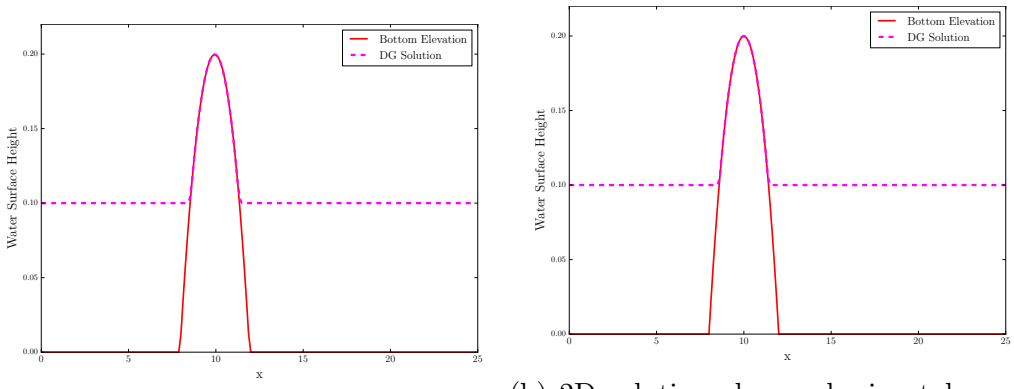
Figure B.4: Short channel with smooth transition and shock

This solution looks slightly under-resolved after the shock near the downstream boundary, where a constant water height is prescribed. However,

it is important to note that the h_{max} we used in this case was 4 times larger than the ones used in Section B.1. The solution does improve as the mesh is refined.

B.3 Lake at rest with an emerged bump

Here, we test not only how well our codes preserve steady state but also how well a dry domain is maintained throughout the simulation. The domain we use is the same as in Section B.1. The initial water height is taken to be 0.1 m, so that the bump emerges from the water. There is no friction and we impose a no-flow boundary condition at $x = 0$ and at $x = 25$. We turn off the slope limiter for the 1-D as well as the 2-D case and set the parameter $H_0 = 1e^{-08}$.



(a) 1D solution

(b) 2D solution along a horizontal cross-section

Figure B.5: Still water at rest around an emerged bump

We obtained Figure B.5 after running the simulation for a thousand seconds. As we can see, for the 1-D as well as the 2-D cases, the steady states

are perfectly preserved and with the wetting and drying treatment, the dry domain remains dry throughout the simulation.

B.4 Transitory solutions

Here, we test how well our codes capture the solution at any given moment in time. Additionally, both of the cases we test have wetting and drying fronts, hence giving us a validation of the wetting and drying treatment that we have incorporated in the codes. This wetting and drying treatment is described in [12].

B.4.1 Dam break on a dry bed

The first case we test is the simple problem of simulating a dam break on a dry bed. For this purpose, we take the following domain:

Domain:

$$0 \text{ m} < x < 25 \text{ m}$$

$$0 \text{ m} < y < 1 \text{ m}$$

Bathymetry:

$$z(x, y) = 0$$

Manning's n:

$$0 \text{ (no friction)}$$

For the 1-D code, we model this region as a 1-D channel with a rectangular cross-section and a uniform width of 1 m. We discretize the domain

with uniform $h = 0.025$ m. For the 2-D code, we have a coarser discretization in the y-direction than in the x-direction. We discretize the 2-D mesh $h_{y,max} = 0.33$ m and $h_{x,max} = 0.025$ m. The initial condition is a modified Riemann problem given by:

Initial Conditions:

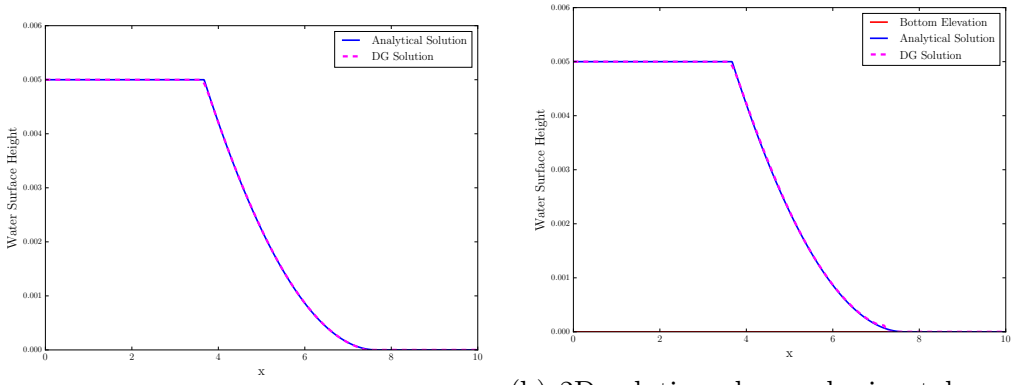
$$\text{Surface elevation:} \quad \zeta = \begin{cases} 0.005 & \text{for } 0 \text{ m} \leq x \leq 5 \\ 0 & \text{for } 5 \text{ m} \leq x \leq 10 \end{cases}$$

$$\text{Flux:} \quad Q^{1D} n = 0 \text{ m}^3/s \qquad \mathbf{Q}^{2D} \cdot n = 0 \text{ m}^2/s$$

Boundary Conditions:

$$\text{@x} = 0: \quad Q^{1D} n = 0 \text{ m}^3/s \qquad \mathbf{Q}^{2D} \cdot \mathbf{n} = 0 \text{ m}^2/s$$

$$\text{@x} = 0: \quad H = 0.005 \text{ m}$$



(a) 1D solution

(b) 2D solution along a horizontal cross-section

Figure B.6: Ideal dam break

For the wetting and drying treatment, we set the parameter H_0 to $1e^{-07}$. Figure B.6 shows the solution six seconds after the dam breaks. As we

can see, the 1-D as well as the 2-D RKDG schemes slightly round the sharp corner at $x = 4$ m. While the 1-D solution is accurate at the wetting front, the wetting front in the 2-D solution is slightly behind the analytical one.

B.4.2 Parabolic Bowl

We use different tests for the 1-D and the 2-D codes in this section. The bathymetry is a parabola for the 1-D case and a radially symmetric paraboloid for the 2-D case. The solutions are periodic and have a moving wetting and drying fronts. This property of the solutions allows us to test how well the wetting and drying treatment preserves mass.

1-D case

Domain:

$$0 \text{ m} < x < L \text{ m}$$

Bathymetry:

$$z(x) = -h_0 \left(\frac{1}{a^2} \left(x - \frac{L}{2} \right)^2 - 1 \right)$$

where, $h_0 = 0.5$ and $L = 4$.

Manning's n:

$$0 \text{ (no friction)}$$

We discretize this domain uniformly with $h = 0.008$ m. For the wetting and drying treatment, we set the parameter H_0 to 0.001. We use a no-flow boundary condition on both ends of the domain and the following initial conditions:

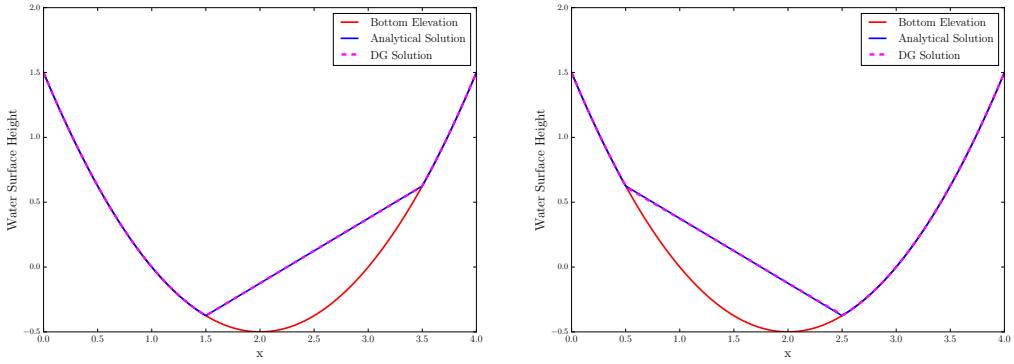
Initial Conditions:

$$\text{Water height: } H = \begin{cases} -h_0 \left(\left(\frac{1}{a} \left(x - \frac{L}{2} \right) + \frac{B}{\sqrt{2gh_0}} \right)^2 - 1 \right) & \text{for } x_1 \leq x \leq x_2 \\ 0 \text{ m} & \text{otherwise} \end{cases}$$

$$\text{Flux: } Q^{1D} n = 0 \text{ m}^2/s$$

where, $B = \frac{\sqrt{2gh_0}}{2a}$ and $a = 1$ m.

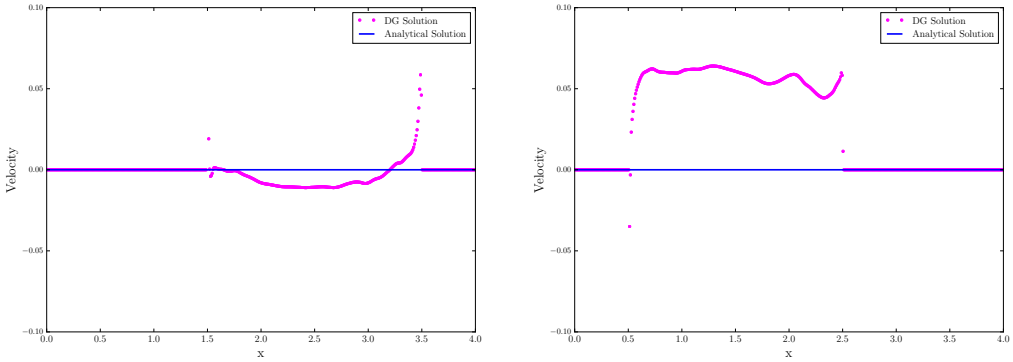
The period of the solution is $T = \frac{2\pi a}{\sqrt{2 * g * h_0}}$. In Figure B.7, we have shown the solutions computed at $t = \frac{1}{2}T$ (B.7a), and at $t = 3T$ (B.7b). As can be seen, the code reproduces the half period solution very accurately. At 3 periods, the RKDG solution is slightly behind the analytical solution. However, the water mass is conserved.



(a) Half a period (b) Three periods

Figure B.7: Periodic solution for water height in a parabolic bowl shaped 1-D domain

In Figure B.8, we have plotted the velocity obtained from our 1-D numerical scheme against the analytical solution at $t = \frac{1}{2}T$ (B.8a) and at $t = 3T$ (B.8b). The analytical velocity is identically zero in the entire domain. The error in the numerical solution for velocity is higher than the error in the numerical solution for water height. The errors in the velocity are small in the interior of the domain, and are higher at the wetting and the drying fronts. However, comparing the error in the velocity at the wetting and the drying fronts at half a period with the error in the velocity at the wetting and the drying fronts at three periods, we can see that these errors do not accumulate in a way that would eventually cause the scheme to be unstable. The maximum of the magnitude of the error at half period as well as three periods is near 0.06.



(a) Half a period (b) Three periods

Figure B.8: Periodic solution for velocity in a parabolic bowl shaped 1-D domain

2-D case

Domain:

$$0 \text{ m} < x < L \text{ m}$$

$$0 \text{ m} < y < L \text{ m}$$

Bathymetry:

$$z(r) = h_0 \left(1 - \frac{r^2}{a^2}\right)$$

Manning's n:

0 (no friction)

The values of the parameters used are: $h_0 = 0.1$ m, $L = 4$, $a = 1$ m and $r = \sqrt{\left(x - \frac{L}{2}\right)^2 + \left(y - \frac{L}{2}\right)^2}$.

We discretize the domain with $h_{x,max} = h_{y,max} = 0.008$ m. For the wetting and drying treatment, we set the parameter H_0 to 0.001. We use a

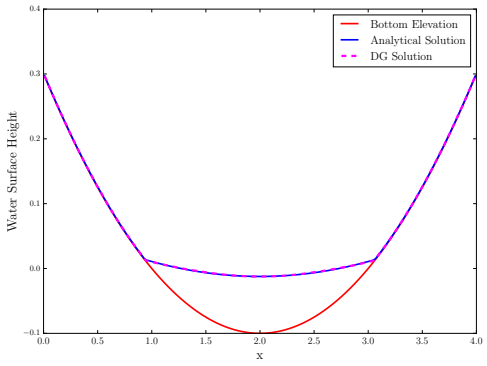
no-flow boundary condition on all four sides of the domain and the following initial conditions:

Initial Conditions:

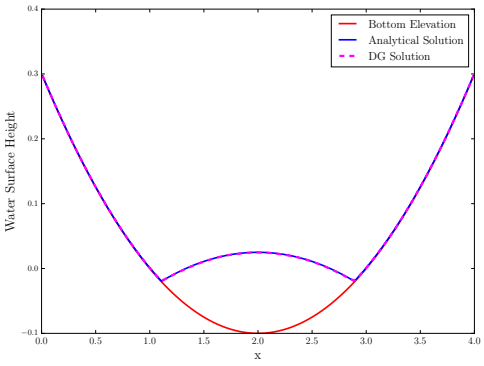
$$\text{Water height:} \quad H(r) = h_0 \left(\frac{\sqrt{1-A^2}}{1-A} - 1 - \frac{r^2}{a^2} \left(\frac{1-A^2}{(1-A)^2} - 1 \right) \right) - z(r)$$

$$\text{Flux:} \quad \mathbf{Q}^{2D} \cdot \mathbf{n} = 0 \text{ m}^2/s$$

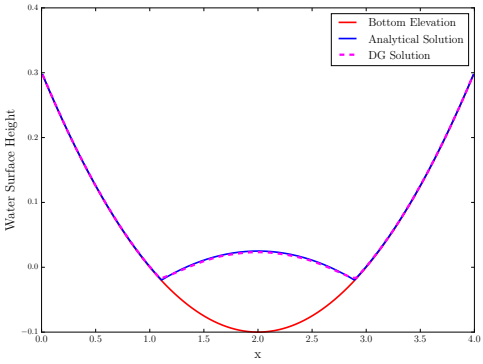
where $A = \frac{a^2 - {r_0}^2}{a^2 + {r_0}^2}$ and $r_0 = 0.8$ m.



(a) Two-thirds of a period



(b) One full period



(c) Three full periods

Figure B.9: Periodic solution along a horizontal cross-section of paraboloid

The period of the solution is $T = \frac{2\pi a}{\sqrt{8 * g * h_0}}$. In Figure B.9, we have

shown the solutions along the cross-section $x = 0$ computed at $t = \frac{2}{3}T$ (B.9a), at $t = T$ (B.9b) and at $t = 3T$ (B.9c). Just like in the 1-D case, our code reproduces the two-thirds period solution and the full period solution very accurately. At 3 periods, the RKDG solution is slightly behind the analytical solution. However, the water mass is conserved.

Bibliography

[1] www.flo-2d.com.

[2] www.mikebydhi.com/products/mike-21.

[3] Hurricane Irene one year later: Storm cost $15.8 in damage from Florida to New York to the Caribbean. http://www.nydailynews.com/new-york/hurricane-irene-year-storm-cost-15-8-damage-florida-new-york-caribbean-artic 1145302, August 2012.

[4] V. Aizinger. *A discontinuous Galerkin Method for two- and three- dimensional shallow-water equations*. PhD thesis, University of Texas at Austin, 2004.

[5] V. Aizinger and C. Dawson. A discontinuous Galerkin method for two-dimensional flow and transport in shallow water. *Advances in Water Resources*, 25(1):67–84, 2002.

[6] Paul D Bates, Richard J Dawson, Jim W Hall, Matthew S Horritt, Robert J Nicholls, Jon Wicks, and Mohamed Ahmed Ali Mohamed Hassan. Simplified two-dimensional numerical modelling of coastal flooding and example applications. *Coastal Engineering*, 52(9):793–810, 2005.

[7] Lance Bode and Thomas A Hardy. Progress and recent developments in storm surge modeling. *Journal of Hydraulic Engineering*, 123(4):315–331, 1997.

[8] O. Bokhove. Flooding and drying in discontinuous Galerkin finite-element discretizations of shallow-water equations. part 1: one dimension. *Journal of scientific computing*, 22(1-3):47–82, 2005.

[9] James D Brown, Tom Spencer, and Iris Moeller. Modeling storm surge flooding of an urban area with particular reference to modeling uncertainties: A case study of canvey island, united kingdom. *Water Resources Research*, 43(6), 2007.

[10] G.W. Brunner. HEC-RAS river analysis system. hydraulic reference manual. version 1.0. Technical report, Hydrologic Engineering Center, U.S. Army Corps of Engineers, 1995.

[11] G.W. Brunner. Combined 1D and 2D modeling with HEC-RAS. `http://www.hec.usace.army.mil/misc/files/ras/Combined_1D_and_2D_Modeling_with_HEC-RAS.pdf`, October 2014.

[12] S. Bunya, E.J. Kubatko, J.J. Westerink, and C.N. Dawson. A wetting and drying treatment for the Runge–Kutta discontinuous Galerkin solution to the shallow water equations. *Computer Methods in Applied Mechanics and Engineering*, 198(17):1548–1562, 2009.

[13] S Bunya, JC Dietrich, JJ Westerink, BA Ebersole, JM Smith, JH Atkinson, R Jensen, DT Resio, RA Luettich, C Dawson, et al. A high-resolution coupled riverine flow, tide, wind, wind wave, and storm surge model for southern louisiana and mississippi. part i: Model development and validation. *Monthly Weather Review*, 138(2):345–377, 2010.

[14] John Charles Butcher. *The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods*. Wiley-Interscience, 1987.

[15] Ven T Chow, David R Maidment, and Larry W Mays. *Applied hydrology*. 1988.

[16] B. Cockburn and C.-W. Shu. The P1–RKDG method for two–dimensional Euler equations of gas dynamics. Technical report, DTIC Document, 1991.

[17] B. Cockburn and C.-W. Shu. The Runge–Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. *Journal of Computational Physics*, 141(2):199–224, 1998.

[18] B. Cockburn and C.-W. Shu. Runge–Kutta discontinuous Galerkin methods for convection–dominated problems. *Journal of scientific computing*, 16(3):173–261, 2001.

[19] B. Cockburn, S. Lin, and C.-W. Shu. TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III:

one–dimensional systems. *Journal of Computational Physics*, 84(1):90–113, 1989.

[20] B. Cockburn, S. Hou, and C.-W. Shu. The Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV: The multidimensional case. *Mathematics of Computation*, 54(190):545–581, 1990.

[21] B. Cockburn, G.E. Karniadakis, and C.-W. Shu. The development of discontinuous Galerkin methods. In B. Cockburn, G.E. Karniadakis, and C.-W. Shu, editors, *Discontinuous Galerkin methods: Theory, Computation and Applications*, volume 11 of *Lecture Notes in Computational Science and Engineering, Part I: Overview*, pages 3–50. Springer, 2000.

[22] Bernardo Cockburn and Clint Dawson. Some extensions of the local discontinuous galerkin method for convection-diffusion equations in multidimensions. In *The proceedings of the conference on the mathematics of finite elements and applications: MAFELAP*, volume X. Elsevier, 2000.

[23] Bernardo Cockburn and Chi-Wang Shu. The local discontinuous galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, 1998.

[24] L. Creel. Ripple effects: Population and coastal regions. `http://www.prb.org/Publications/Reports/2003/RippleEffectsPopulationandCoastalRegions.aspx`, September 2003.

[25] J.A. Cunge, Jr. Holly, F.M., and A. Verwey. *Practical aspects of computational river hydraulics*. Pitman publishing, 1980.

[26] C. Dawson, E.J. Kubatko, J.J. Westerink, C. Trahan, C. Mirabito, C. Michoski, and N. Panda. Discontinuous Galerkin methods for modeling hurricane storm surge. *Advances in Water Resources*, 34:1165–1176, 2011.

[27] O. Delestre, C. Lucas, P. Ksinant, F. Darboux, C. Laguerre, T. Vo, F. James, and S. Cordier. SWASHES: a compilation of shallow water analytic solutions for hydraulic and environmental studies. *International Journal for Numerical Methods in Fluids*, 72(3):269–300, 2013.

[28] JC Dietrich, S Bunya, JJ Westerink, BA Ebersole, JM Smith, JH Atkinson, R Jensen, DT Resio, RA Luettich, C Dawson, et al. A high-resolution coupled riverine flow, tide, wind, wind wave, and storm surge model for southern louisiana and mississippi. part ii: Synoptic description and analysis of hurricanes katrina and rita. *Monthly Weather Review*, 138(2): 378–404, 2010.

[29] JC Dietrich, M Zijlema, JJ Westerink, LH Holthuijsen, C Dawson, Richard A Luettich, RE Jensen, JM Smith, GS Stelling, and GW Stone. Modeling hurricane waves and storm surge using integrally-coupled, scalable computations. *Coastal Engineering*, 58(1):45–65, 2011.

[30] Shishir K Dube, Tad S Murty, Jesse C Feyen, B Harper, J Bales, and Saud Amer. Storm surge modeling and applications in coastal areas. *Global*

perspectives on tropical cyclones: from science to mitigation, pages 363–406, 2010.

[31] C. Eskilsson and S.J. Sherwin. A triangular spectral/hp discontinuous Galerkin method for modelling 2d shallow water equations. *International Journal for Numerical Methods in Fluids*, 45(6):605–623, 2004.

[32] Cristina Forbes, Richard A Luettich Jr, Craig A Mattocks, and Joannes J Westerink. A retrospective evaluation of the storm surge produced by hurricane gustav (2008): Forecast and hindcast results. *Weather and Forecasting*, 25(6):1577–1602, 2010.

[33] T Graham Freeman. Calculating catchment area with divergent flow based on a regular grid. *Computers & Geosciences*, 17(3):413–422, 1991.

[34] R. Gandham, D.S. Medina, and T. Warburton. Gpu accelerated discontinuous Galerkin methods for shallow water equations. *arXiv preprint arXiv:1403.1661*, 2014.

[35] M.P. Garcia-Navarro and J.M. Savirón. Numerical simulation of unsteady flow at open channel junctions. *Journal of Hydraulic Research*, 30(5):595–609, 1992.

[36] P. Garcia-Navarro, P. Brufau, J. Burguete, and J. Murillo. The shallow water equations: An example of hyperbolic system. *Monografķas de la Real Academia de Ciencias de Zaragoza*, 31, 2008.

[37] D. Gilles and M. Moore. Review of hydraulic flood modeling software used in belgium, the netherlands, and the united kingdom. Technical report, International perspectives in water resource management, IIHR–Hydroscience and Engineering, August 2010.

[38] F.X. Giraldo and M. Restelli. High-order semi-implicit time-integrators for a triangular discontinuous Galerkin oceanic shallow water model. *International journal for numerical methods in fluids*, 63(9):1077–1102, 2010.

[39] F.X. Giraldo and T. Warburton. A high-order triangular discontinuous Galerkin oceanic shallow water model. *International journal for numerical methods in fluids*, 56(7):899–925, 2008.

[40] F.X. Giraldo, J.S. Hesthaven, and T. Warburton. Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations. *Journal of Computational Physics*, 181(2):499–525, 2002.

[41] Bob Glahn, Arthur Taylor, Nicole Kurkowski, and Wilson A Shaffer. The role of the slosh model in national weather service storm surge forecasting. *National Weather Digest*, 33(1):3–14, 2009.

[42] O. Gourgue, R. Comblen, J. Lambrechts, T. Kärnä, V. Legat, and E. Deleersnijder. A flux-limiting wetting–drying method for finite-element shallow-water models, with application to the scheldt estuary. *Advances in Water Resources*, 32(12):1726–1739, 2009.

[43] K. Havnø, M.N. Madsen, J. Dørge, V.P. Singh, et al. MIKE 11 – a generalized river modelling package. *Computer models of watershed hydrology.*, pages 733–782, 1995.

[44] Jan S Hesthaven and Tim Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.

[45] Mark E Hope, Joannes J Westerink, Andrew B Kennedy, PC Kerr, JC Dietrich, C Dawson, Christopher J Bender, JM Smith, Robert E Jensen, Marcel Zijlema, et al. Hindcast and validation of hurricane ike (2008) waves, forerunner, and storm surge. *Journal of Geophysical Research: Oceans*, 118(9):4424–4460, 2013.

[46] N.M. Hunter, P.D. Bates, M.S. Horritt, and M.D. Wilson. Simple spatially-distributed models for predicting flood inundation: a review. *Geomorphology*, 90(3):208–225, 2007.

[47] Chester P Jelesnianski, Jye Chen, and Wilson A Shaffer. Slosh: Sea, lake, and overland surges from hurricanes. *NOAA Technical Report NWS 48*, 1992.

[48] Susan K Jenson and Julia O Domingue. Extracting topographic structure from digital elevation data for geographic information system analysis. *Photogrammetric engineering and remote sensing*, 54(11):1593–1600, 1988.

[49] Cevza Melek Kazezyılmaz-Alhan, Miguel A Medina, and Prasada Rao. On numerical modeling of overland flow. *Applied Mathematics and Computation*, 166(3):724–740, 2005.

[50] G. Kesserwani and Q. Liang. A discontinuous Galerkin algorithm for the two-dimensional shallow water equations. *Computer Methods in Applied Mechanics and Engineering*, 199(49):3356–3368, 2010.

[51] G. Kesserwani, R. Ghostine, J. Vazquez, R. Mosé, M. Abdallah, and A. Ghenaim. Simulation of subcritical flow at open-channel junction. *Advances in Water Resources*, 31(2):287–297, 2008.

[52] G. Kesserwani, Q. Liang, J. Vazquez, and R. Mosé. Well-balancing issues related to the rkdg2 scheme for the shallow water equations. *International journal for numerical methods in fluids*, 62(4):428–448, 2010.

[53] E.J. Kubatko, J.J. Westerink, and C. Dawson. *hp* discontinuous Galerkin methods for advection dominated problems in shallow water flow. *Computer Methods in Applied Mechanics and Engineering*, 196(1):437–451, 2006.

[54] E.J. Kubatko, S. Bunya, C. Dawson, and J.J. Westerink. Dynamic *p*-adaptive Runge–Kutta discontinuous Galerkin methods for the shallow water equations. *Computer Methods in Applied Mechanics and Engineering*, 198(21):1766–1774, 2009.

[55] Ethan J Kubatko, Joannes J Westerink, and Clint Dawson. hp discontinuous galerkin methods for advection dominated problems in shallow water flow. *Computer Methods in Applied Mechanics and Engineering*, 196(1):437–451, 2006.

[56] Ethan J Kubatko, Shintaro Bunya, Clint Dawson, Joannes J Westerink, and Chris Mirabito. A performance comparison of continuous and discontinuous finite element shallow water models. *Journal of Scientific Computing*, 40(1-3):315–339, 2009.

[57] S.N. Kuiry, D. Sen, and P.D. Bates. Coupled 1d–quasi-2d flood inundation model with unstructured grids. *Journal of Hydraulic Engineering*, 136(8): 493–506, 2010.

[58] JP Lapidez, J Tablazon, L Dasallas, LA Gonzalo, KM Cabacaba, MMA Ramos, JK Suarez, J Santiago, AMF Lagmay, and V Malano. Identification of storm surge vulnerable areas in the philippines through the simulation of typhoon haiyan-induced storm surge levels over historical storm tracks. *Nat. Hazards Earth Syst. Sci*, 15:1473–1481, 2015.

[59] NL Lea. An aspect driven kinematic routing algorithm. *Overland flow: hydraulics and erosion mechanics*, 147:175, 1992.

[60] J. Leandro, A.S. Chen, S. Djordjević, and D.A. Savić. Comparison of 1d/1d and 1d/2d coupled (sewer/surface) hydraulic models for urban flood simulation. *Journal of hydraulic engineering*, 135(6):495–504, 2009.

[61] RA Leuttich Jr, JJ Westerink, and NW Scheffner. Adcirc: An advanced three-dimensional circulation model for shelves, coasts, and esturarie. i: Theory and methodology of adcirc-2ddi and adcirc-3dl. Technical report, DRP Tech. Rep. 1, US Army Corps of Engineers, Waterways Experiment Station, Vicksburg, MS, 1992.

[62] R.J. LeVeque. *Numerical methods for conservation laws*. Birkhäuser, Basel, 2nd edition, 1992.

[63] H. Li and R. Liu. The discontinuous Galerkin finite element method for the 2d shallow water equations. *Mathematics and computers in simulation*, 56(3):223–233, 2001.

[64] A. Mahjoob and R. Ghiassi. Application of a coupling algorithm for the simulation of flow and pollution in open channels. *World Applied Sciences Journal*, 12(4):446–459, 2011.

[65] K. Mahmood, V.M. Yevjevich, and W.A. Miller. *Unsteady flow in open channels*, volume 1. Water Resources Publications Fort Collins, Colorado, USA, 1975.

[66] O. Mark, S. Weesakul, C. Apirumanekul, S.B. Aroonnet, and S. Djordjević. Potential and limitations of 1d modelling of urban flooding. *Journal of Hydrology*, 299(3):284–299, 2004.

[67] David G Morris and Richard G Heerdegen. Automatically derived catch-

ment boundaries and channel networks and their hydrological applications. *Geomorphology*, 1(2):131–141, 1988.

[68] R. Muñoz-Carpena, C.T. Miller, and J.E. Parsons. A quadratic petrov-galerkin solution for kinematic wave overland flow. *Water resources research*, 29(8):2615–2627, 1993.

[69] National Oceanic National Hurricane Center and Atmospheric Administration. Sea, lake and overland surges from hurricanes (SLOSH). http://www.nhc.noaa.gov/surge/slosh.php.

[70] P. Neupane and C. Dawson. A discontinuous Galerkin method for modeling flow in networks of channels. *Advances in Water Resources*, 79:61–79, 2015.

[71] John F O'Callaghan and David M Mark. The extraction of drainage networks from digital elevation data. *Computer vision, graphics, and image processing*, 28(3):323–344, 1984.

[72] Pamela Probst, Giovanni Franchello, et al. *Global storm surge forecast and inundation modeling*. Joint Research Centre, European Commission, 2012.

[73] PFBJ Quinn, K Beven, Pierre Chevallier, and Olivier Planchon. The prediction of hillslope flow paths for distributed hydrological modelling using digital terrain models. *Hydrological processes*, 5(1):59–79, 1991.

[74] J Tyler Ray, Z Fang, and PB Bedient. Assessment of flood risk due to storm surge in coastal bayous using dynamic hydraulic modeling. In *World Environmental and Water Resources Congress*, pages 1–10, 2009.

[75] W.H. Reed and T.R. Hill. Triangular meshes methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[76] J.F. Remacle, S.S. Frazao, X. Li, and M.S. Shephard. An adaptive discretization of shallow-water equations based on discontinuous Galerkin methods. *International journal for numerical methods in fluids*, 52(8): 903–923, 2006.

[77] P.L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43(2):357–372, 1981.

[78] A. Rohatgi. Webplotdigitizer. `http://arohatgi.info/WebPlotDigitizer`.

[79] M. Rungo and K.W. Olesen. Combined 1-and 2-dimensional flood modelling. In *Fourth Iranian Hydraulic Conference*, pages 21–23, 2003.

[80] The USGS Water Science School. What is hydrology and what do hydrologists do? `water.usgs.gov/edu/hdyrology.html`.

[81] D. Schwanenberg and M. Harms. Discontinuous Galerkin finite-element method for transcritical two-dimensional shallow water flows. *Journal of Hydraulic Engineering*, 130(5):412–421, 2004.

[82] D. Schwanenberg and J. Köngeter. A discontinuous Galerkin method for the shallow water equations with source terms. In Cockburn B, Karniadakis GE, and Shu C-W, editors, *Discontinuous Galerkin methods: Theory, Computation and Applications*, Lecture Notes in Computational Science and Engineering, Part III, pages 419–424. Springer, 2000.

[83] S.D. Seyoum, Z. Vojinovic, R.K. Price, and S. Weesakul. Coupled 1d and noninertia 2d flood inundation model for simulation of urban flooding. *Journal of Hydraulic Engineering*, 138(1):23–34, 2012.

[84] C. Sherman. Rio grande flood called unique in timing, scale. `https://www.victoriaadvocate.com/news/2010/jul/22/bc-tx-rio-grande-flood/`, July 2010.

[85] LeRoy K Sherman. Streamflow from rainfall by the unit-graph method. *Eng. News Record*, 108:501–505, 1932.

[86] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439–471, 1988.

[87] Vijay P Singh and David A Woolhiser. Mathematical modeling of watershed hydrology. *Journal of hydrologic engineering*, 7(4):270–292, 2002.

[88] VP Singh. Is hydrology kinematic? *Hydrological processes*, 16(3):667–716, 2002.

[89] David G Tarboton. A new method for the determination of flow directions and upslope areas in grid digital elevation models. *Water resources research*, 33(2):309–319, 1997.

[90] E Tromble, R Kolar, K Dresback, and R Luettich. River flux boundary considerations in a coupled hydrologic-hydrodynamic modeling system. In *Estuarine and Coastal Modeling, Proceedings of the Twelfth International Conference. ASCE*, pages 510–527, 2012.

[91] EM Tromble, RL Kolar, KM Dresback, Y Hong, BE Vieux, RA Luettich, JJ Gourley, KE Kelleher, and S Van Cooten. Aspects of coupled hydrologic–hydrodynamic modeling for coastal flood inundation. In *Estuarine and Coastal Modeling: Proceedings of the Eleventh International Conference, Spaulding ML (ed.). American Society of Civil Engineers Publications: Reston, Virginia*, pages 724–743, 2010.

[92] Baxter E Vieux. *Distributed hydrologic modeling using GIS*. Springer, 2001.

[93] L.J Weber, E.D. Schumate, and N. Mawer. Experiments on flow at a 90° open-channel junction. *Journal of Hydraulic Engineering*, 127(5):340–350, 2001.

[94] M.G.F. Werner. A comparison of flood extent modelling approaches through constraining uncertainties on gauge data. *Hydrology and Earth System Sciences*, 8(6):1141–1152, 2004.

[95] Joannes J Westerink, Richard A Luettich, Jesse C Feyen, John H Atkinson, Clint Dawson, Hugh J Roberts, Mark D Powell, Jason P Dunion, Ethan J Kubatko, and Hasan Pourtaheri. A basin-to channel-scale unstructured grid hurricane storm surge model applied to southern louisiana. *Monthly Weather Review*, 136(3):833–864, 2008.

[96] D.A. Woolhiser and J.A. Liggett. Unsteady, one-dimensional flow over a planethe rising hydrograph. *Water Resources Research*, 3(3):753–771, 1967.

[97] Y. Xing and C.-W. Shu. High order well-balanced finite volume weno schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms. *Journal of Computational Physics*, 214(2): 567–598, 2006.

[98] Y. Xing and X. Zhang. Positivity-preserving well-balanced discontinuous galerkin methods for the shallow water equations on unstructured triangular meshes. *Journal of Scientific Computing*, 57(1):19–41, 2013.